UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE INGENIERIA

PROGRAMA MONITOR PARA COMUNICACION SERIE Y SISTEMA DE DESARROLLO POR DESCARGA PARA UN SISTEMA BASADO EN EL MICROPROCESADOR Z-80.

Tesis Profesional

Que para obtener el Título de

INGENIERO MECANICO ELECTRICISTA

Area de Circuitos Eléctricos y Electrónicos

p r e s e n t a

GERARDO ENRIQUE NUÑEZ PITTY

México, D. F. 1983

DEDICATORIA

A MIS PADRES Y HERMANOS
POR SU APOYO DURANTE MIS ESTUDIOS

A IBETT Y SUHHEL POR SU APOYO
CONFIANZA Y ESPERA.

AGRADECIMIENTOS

AL ING. PEDRO JOSELEVICH COHEN POR SU GUIA EN LA ELABORACION DE ESTE TRABAJO, AL ING. LUIS CORDERO B. POR LA AYUDA QUE PRESTO AL PROYECTO, AL COMPAÑERO LUIS BARBA B. POR LA COOPERACION PARA LA EDICION DEL TRABAJO, A TODOS LOS AMIGOS Y COMPAÑEROS Y EN GENERAL A TODAS LAS PERSONAS QUE ME HAN APOYADO DURANTE MIS ESTUDIOS Y SIN LAS CUALES NO HABRIA SIDO POSIBLE LA CULMINACION DE ESTOS.

CONTENIDO

INTRODUCCION

En la actualidad la utilizaciòn masiva de los Microprocesadores debido a su veloz desarrollo y economia, hace indispensable que se cuente con una capacitaciòn adecuada para el diseño basado en este tipo de dispositivo.

Por esta razon se ha realizado este trabajo en el cual se describe el diseño de un programa monitor para un sistema basado en un mP Z80. Si bien es cierto que debemos conocer la arquitectura de un mP para poder hacer algun diseño con éste, es importante saber manejar en forma eficiente los equipos que tenemos a nuestro alcance para poder desarrollar un proyecto con mP.

En este trabajo no solo se tuvo como objetivo desarrollar un Programa Monitor, sino también, conocer un sistema de desarrollo y hacer una pequeña expansiòn de éste. En este proyecto se contò con la ayuda de un sistema Cromenco S2 para el desarrollo del Programa Monitor, con un equipo de entrenamiento para el mP Z80 para programaciòn de memorias e interconexiòn con el sistema Cromenco.

El objetivo principal para utilizar el equipo de entrenamiento para Z80, fué el de probar el programa monitor fuera del equipo Cromenco, y hacer una interconexiòn entre los dos equipos para poder traspasar programas desarrollados en el equipo Cromenco al otro equipo y allí poder pasarlo a

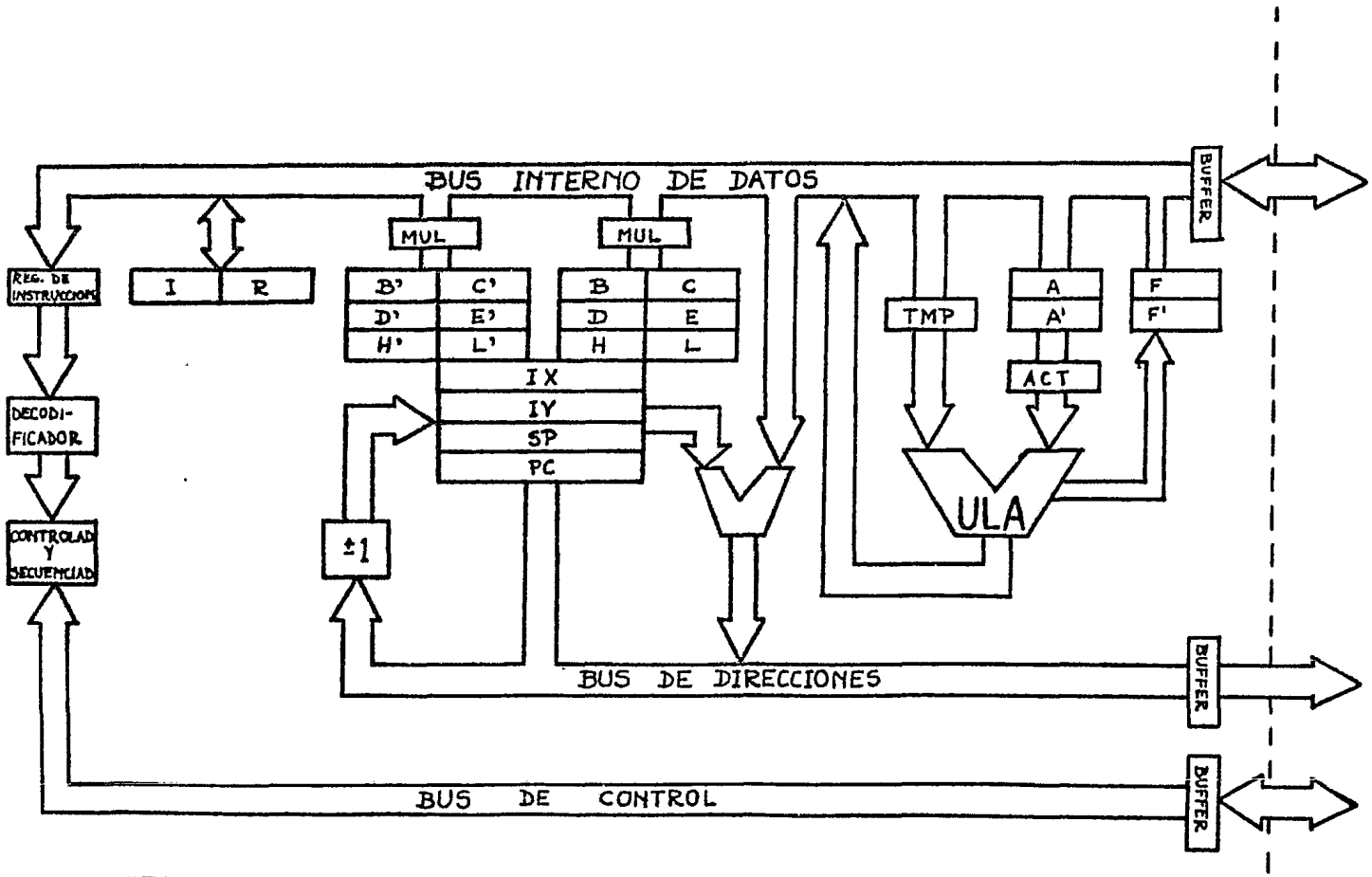
una memoria EPROM o probar programas en tiempo real.

El Programa monitor fue diseñado en forma general para que pueda correr en diversos sistemas y bajo diferentes ambientes del microprocesador, con solo hacer pequeños cambios en el programa. La ùnica restricciòn es que el equipo deberà contar con un puerto de entrada y salida en serie. Para el caso particular del equipo de entrenamiento para el mP Z80, se le agregò un puerto en serie para comunicaciòn con una terminal y con el sistema Cromenco, ademàs se hizo un comando particular para el equipo en el programa monitor para poder Programar memorias EPROM.

ARQUITECTURA DEL MICROPROCESADOR Z80

El mP Z80 fue diseñado para sustituir al popular mP 8080 de Intel, ofreciendo una compatibilidad total a nivel de software con este mP, eliminando varios de los circuitos externos que utiliza el mP 8080 y brindando varias ventajas tanto a nivel de software como a nivel de hardware.

En el diagrama podemos observar la arquitectura del mP Z80; al extremo derecho observamos la Unidad Logica Arimetica (ULA), donde se ejecutan las instrucciones que realizan operaciones lògicas o arimeticas. La ULA esta conectada al bus externo de datos y a todos los registros internos de la CPU. A cada entrada de la ULA, encontramos registros temporales; uno para el acumulador y otro para el otro operando. El registro F mostrado a la derecha del acumulador contiene las banderas que son condicionadas por el resultado de las operaciones realizadas en la ULA. Para el acumulador (A) y las banderas (F) existen dos registros primos A' y F' respectivamente los cuales pueden ser intercambiados por A y F pero no utilizados simultàneamente. Luego encontramos un conjunto de registros para propòsito general, éstos son: B,C,D,E,H y L los cuales tienen una longitud de palabra de 8 bits, para estos registros existe otro grupo de registros por el cual puede ser intercambiado (B',C',D',E',H' Y L').



ORGANIZACION INTERNA DEL Z80.

Al igual que para el àcumulador y las banderas solo puede estar activo un grupo de estos registros a la vez. En la parte superior de los registros se ve un pequeño rectàngulo con el nombre MUL, este representa el multiplexor para escoger entre los dos grupos de registros. Todos estos registros pueden ser utilizados en pares (BC, DE Y HL) como registros de 16 bits para apuntadores de memoria. Los cuatro registros de la parte inferior son utilizados exclusivamente para direccionamiento de memoria. Como en cualquier mP contamos con un contador de programa, el cual posee la direcciòn de 16 bits donde se encuentra la siguiente instrucciòn a ejecutarse. El contador de programa se incrementa automàticamenete una vez que la instrucciòn ha sido traida de memoria. El apuntador de pila o SP nos señala la direcciòn actual del tope de la pila localizada en cualquier parte de la memoria RAM. Los registros ìndices (IX e IY) tambien de 16 bits poseen la direcciòn base que se utiliza para el direccionamiento indexado. En este modo un registro ìndice apunta a la base de una regiòn de memoria de donde se traeràan datos o a donde se enviaràn datos. Un byte adicional es incluido en la instrucciòn para especificar un desplazamiento a partir de la base, el desplazamiento es expresado en complemento a dos y es agregado a la base a través de la pequeña ULA mostrada en la figura. Finalmente el pequeño rectàngulo con +/- ,que aparece a la izquierda de los registros de 16 bits sirve para incrementar o decrementar

éstos automàticamente. Casi al extremo izquierdo se muestran dos registros muy particulares de este mP, el registro I que guarda la parte alta de un vector de interrupciones y el registro R que es un registro especial para el refrescamiento de memorias dinàmicas. Al extremo izquierdo de la figura se puede observar el registro de instrucciòn, que contendrà la instrucciòn que serà ejecutada; esta instrucciòn serà decodificada por el decodificador mostrado, el que enviarà señales al secuenciador-controlador, el cual ejecutarà la instrucciòn.

Interrupciones en el Microprocesador Z80.

Una interrupciòn es una señal que se envìa al mP en cualquier tiempo solicitando servicio de éste y es Asincrona al programa en ejecuciòn. Una interrupciòn se puede generar en cualquier tiempo y por lo general suspende la ejecuciòn del programa.

El Z80 provee al usuario de tres mecanismos diferentes de interrupciòn: Solicitud de Bus, Interrupciòn no mascarable y la Interrupciòn mascarable.

Solicitud de Bus.

Es el mecanismo de mas alta prioridad de interrupciòn del Z80. Como regla general ninguna señal puede ser sensada por el mP hasta que finalice el ciclo de maquina actual. En el caso de la interrupciòn no mascarable y la mascarable no son sensadas hasta que a instrucciòn corriente halla finalizado. La Solicitud de Bus solo necesita que el ciclo de maquina sea finalizado ésta interrupciòn es utilizada para realizar un DMA (Acceso Directo de Memoria).

Interrupciòn no Mascarable.

Esta interrupciòn no puede ser deshabilitada por programa (de aqui su nombre). La interrupciòn no mascarable hace que el Cont. de Prog. se guarde en la pila y se realize un salto a la localidad 0066Hex., en ésta localidad se debe encontrar la rutina de servicio a la interrupciòn.

Interrupciòn Mascarable.

Esta interrupciòn puede operar en uno de tres modos diferentes. Esta es una caracterìstica del Z80. Para programar el modo de operaciòn se utiliza la instrucciòn M0,M1 o M2 con las cuales programamos al mP para el modo correspondiente. En cualquiera de los tres modos podemos utilizar las instrucciones DI o EI con las cuales deshabilitamos y habilitamos interrupciones respectivamente.

Modo de interrupcion 0.

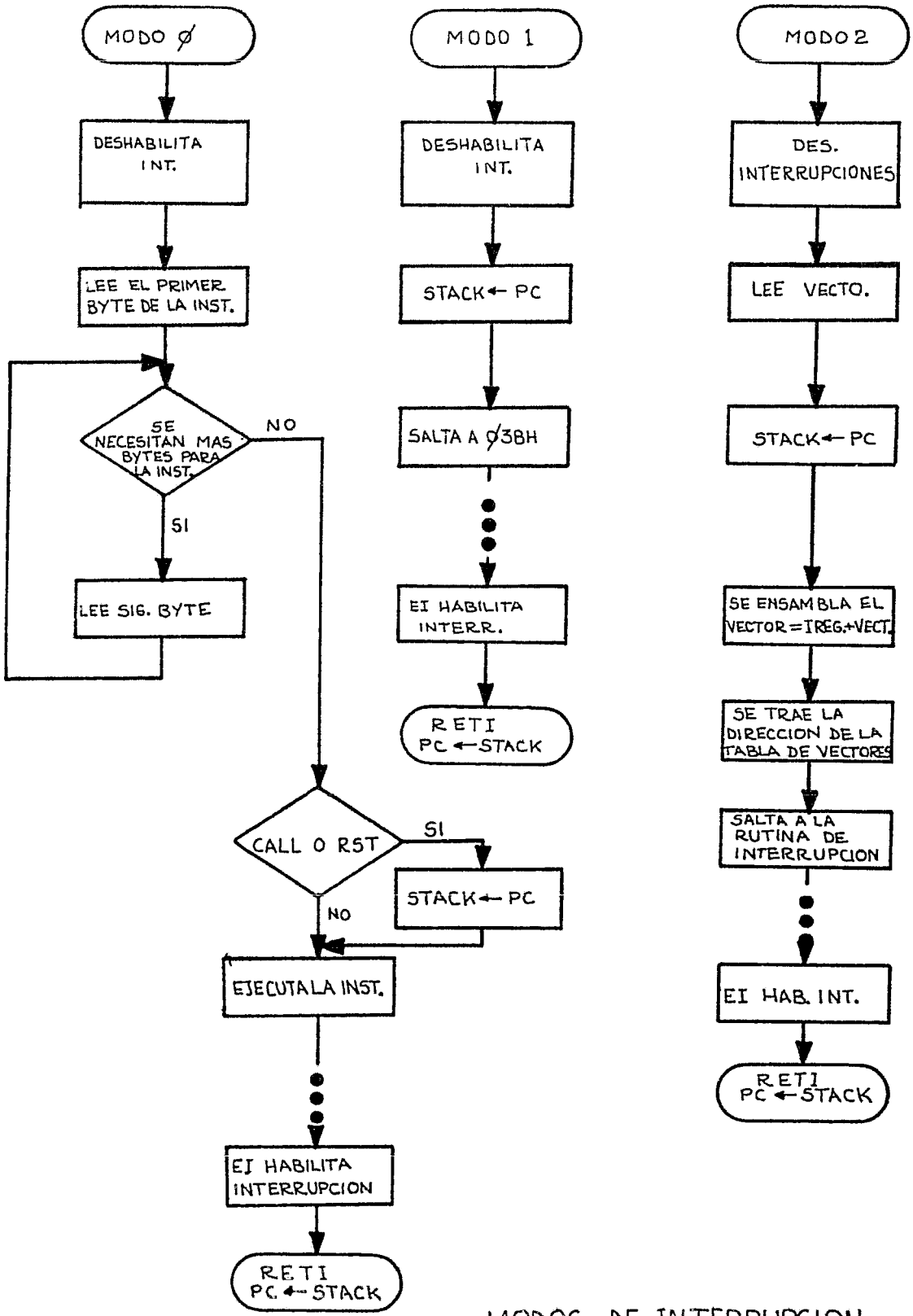
Este modo de interrupciòn es identico al del mP 8080 de Intel, cuando la interrupciòn es sensada y se opera en éste modo el Z80 responde a la interrupciòn con una señal de "reconocimiento de interrupciòn" . Con lo que el periferico que interrumpio al mP deberà responer colocando una instrucciòn en el bus de datos, por lo general se coloca una intrucciòn RST o CALL.

Modo de interrupciòn 1.

Esta interrupciòn es similar a la interrupciòn no mascarable ya que cuando se sensa la señal de interrupciòn se guarda el contador de programa en la pila y se hace un salto incondicional a la localidad 0038Hex. Esto hace que no se necesite hardware externo pero tiene inconveniencias si hay varios dispositivos que generen interrupciones.

Modo de interrupciòn 2.

Este modo de interrupciòn es exclusivo del mP Z80. En este modo se cuenta con un vector de interrupciòn el cual apunta a las localidades donde se encuentra la direcciòn de la rutina de atenciòn a la interrupciòn. El byte màs significativo del vector de interrupciones se encuntra en el registro I del Z80 y el byte menos significativo de éste lo debe proveer el dispositvo que interrumpa cuando la señal de interrupciòn es reconocida.



MODOS DE INTERRUPCION NO MASCARABLE DEL Z80.

Descripcion del programa Monitor ZBUG

del equipo Starter Kit Z80

El programa Zbug es un monitor de 2K bytes diseñado por la Cia. S.D. Systems para que el usuario del equipo Starter Kit Z80 pueda desarrollar y correr programas en lenguaje de màquina Z80. Este monitor utiliza un teclado Hexadecimal para la entrada de datos y un despliegue de seis displays de siete segmentos para la lectura de datos del Starter Kit Z80. El monitor incluye programas de salvado y carga de programas en cassets de audio; puede colocar varios puntos de ruptura en programas de usuario ademàs de correr programas instrucciòn por instrucciòn (paso a paso), poder revisar y cambiar el contenido de memoria y registros internos de mP Z80.

Reset

El equipo cuenta con un interruptor que produce la reinicializaciòn del sistema, lo que hace que el contador de programa se cargue con el valor 0000H el cual indica la localidad de memoria donde empezarà el programa monitor Zbug. El primer paso del monitor es inicializar el apuntador de pila del mP Z80 con el valor 23C0h, inicializar las variables de memoria RAM y si el interruptor S3 està en posiciòn "Monitor Restart" colocarà un guiòn en los siete segmentos mas a la izquierda del desplegado, en señal de que estamos

bajo el control del Zbug. En caso que S3 esté en la posiciòn "Prom1 Restart" el monitor enviarà el control del programa a la localidad 800H. Para cualquier posiciòn que se coloque S3 el monitor harà la inicializaciòn mencionada, deshabilitarà interrupciones y escogerà el modo de interrupciones 0 del mP Z80 (identico al modo de interrupciones del mP 8080 de Intel).

Comandos del Monitor ZBUG.

Salvar a Casset: Salva en cassets de audio programas de memoria RAM o ROM, utilizando las normas de formato de Kansas City. Para este comando se dan como paràmetros la localidad donde empieza el bloque de memoria que se salvarà y la localidad final. El primer paràmetro se introduce en las localidades 23C0H y 23C1H, el segundo paràmetro ira a las localidades 23C2H y 23C3H.

Carga:Este comando carga programas salvados en memoria y los colocarà en las localidades de donde originalmente fueron salvados.

Desplegado de registros:este comando despliega el contenido de los registros que deseamos observar. Para esto debemos de teclear el registro deseado y luego la tecla REG EXAM o REG EXAM' para los registros primos.

Examen de Memoria:Este comando nos permite ver el contenido de memoria RAM o ROM además de poder modificar el contenido de memoria RAM. Para lo anterior tecleamos la localidad de memoria (los ultimos cuatro digitos tecleados) y oprimimos la tecla MEM EXAM.

Examen de Puertos:Mediante este comando se pueden examinar y cambiar contenidos de puertos. Para esto se teclean dos digitos hexadecimales y luego la tecla PORT EXAM.

Puntos de Ruptura:Tecleando cuatro digitos y luego la tecla BREAKPOINT podremos colocar un punto de ruptura en nuestros programas. Estos puntos de ruptura se "hacen" substituyendo temporalmente el codigo de operaciòn de la localidad dada por el valor 0CFH el cual corresponde a la instrucciòn Restart 08 del mP Z80 la cual salvarà el contador de programa y lo reemplazarà con el valor 0008H donde hay una rutina que salva los valores que tienen los registros del mP Z80 y pasa el control al monitor Zbug nuevamente. El monitor mantiene control de los puntos de ruptura que colocamos en un programa (que pueden ser hasta cinco).

Instrucciòn por instrucciòn:Este comando nos permite correr

un programa instrucciòn por instrucciòn. Para esto debemos inicializar el contador de programa con la direcciòn de memoria de nuestro programa y luego teclear la tecla SINGLE STEP la cual harà que ejecute una instrucciòn; se salven los registros y regrese el control al Zbug. Esto se logra de la siguiente manera: cuando se oprime la tecla SINGLE STEP se restauran los valores de los registros del mP Z80 que estàn salvados en memoria y se programa el CTC (Circuito Contador y Temporizador Z80) para que genere una interrupciòn no mascarable, esta interrupciòn esta calculada que se produzca durante la ejecuciòn de la primera instrucciòn del pograma, de esta manera solo se ejecuta una instrucciòn. La rutina de interrupciòn salva los registros en memoria RAM y regresa el control al Zbug.

Monitor: este comando tambien genera una interrupciòn no mascarable la cual salva el contenido de los registros del Z80 y regresa el control al Zbug.

Ejecuciòn:este comando restaura el contenido de los registros y ejecuta un programa desde el valor dado por el usuario o desde el valor que tenia el contador de programa.

Programaciòn:este comando nos permite programar una memoria EPROM 2758 o 2716 5volts. Para esto se toma el numero de

bytes que sera movido desde RAM (empezando desde la localidad 2000H) a la EPROM (en la localidad 1000H) y luego la tecla PROGRAM para que sea programada la EPROM. Para poder utilizar este comando se necesita una fuente de voltaje adicional de 25volts.

Proximo:este comando nos permite ver el contenido de la proxima localidad de memoria si estamos en el comando de examen de memoria o el proximo puerto si estamos en el comando de examen de puertos.

Arimetica Hexadecimal:Esta rutina del monitor nos permite calcular los desplazamientos para saltos relativos.

Ademàs de los comandos mencionados el monitor Zbug posee rutinas que pueden ser utilizadas por el usuario.

PROGRAMA MONITOR PARA UTILIZAR TERMINAL

Este Programa Monitor fue desarrollado pensando en ofrecer las facilidades comunes de un programa de este tipo además de brindar opciones particulares al sistema. Para esto se tomò en cuenta las utilidades con que cuenta el mP Z80, por supuesto que esto en cuanto al conjunto de intrucciones con que cuenta. El programa puede ser dividido en cuatro grupos de codigo principalmente; tres de ellos forman el programa en si y el ultimo lo compone el conjunto de datos requeridos por el programa. Para poder desarrollar el programa se adaptò un Usart al Equipo de entrenamiento Starter Kit Z80; además, se utilizò el circuito CTC de la familia del mP Z80 para que generarà la señal de reloj para el baudaje del sistema. Los tres primeros grupos de còdigo del programa lo componen: Inicializaciòn de periféricos y variables, Rutinas de uso general y particular; por ultimo tenemos los comandos con que cuenta el programa.

Como se puede apreciar en el diagrama de flujo, el programa Monitor es bastante sencillo en su aspecto general. Aquì podemos ver que luego de realizar la inicializaciòn el programa solo espera un caracter el cual debe corresponder a un Comando y de no ser asi el control del programa saltara a una rutina de error y de allì hasta el reconocimiento del

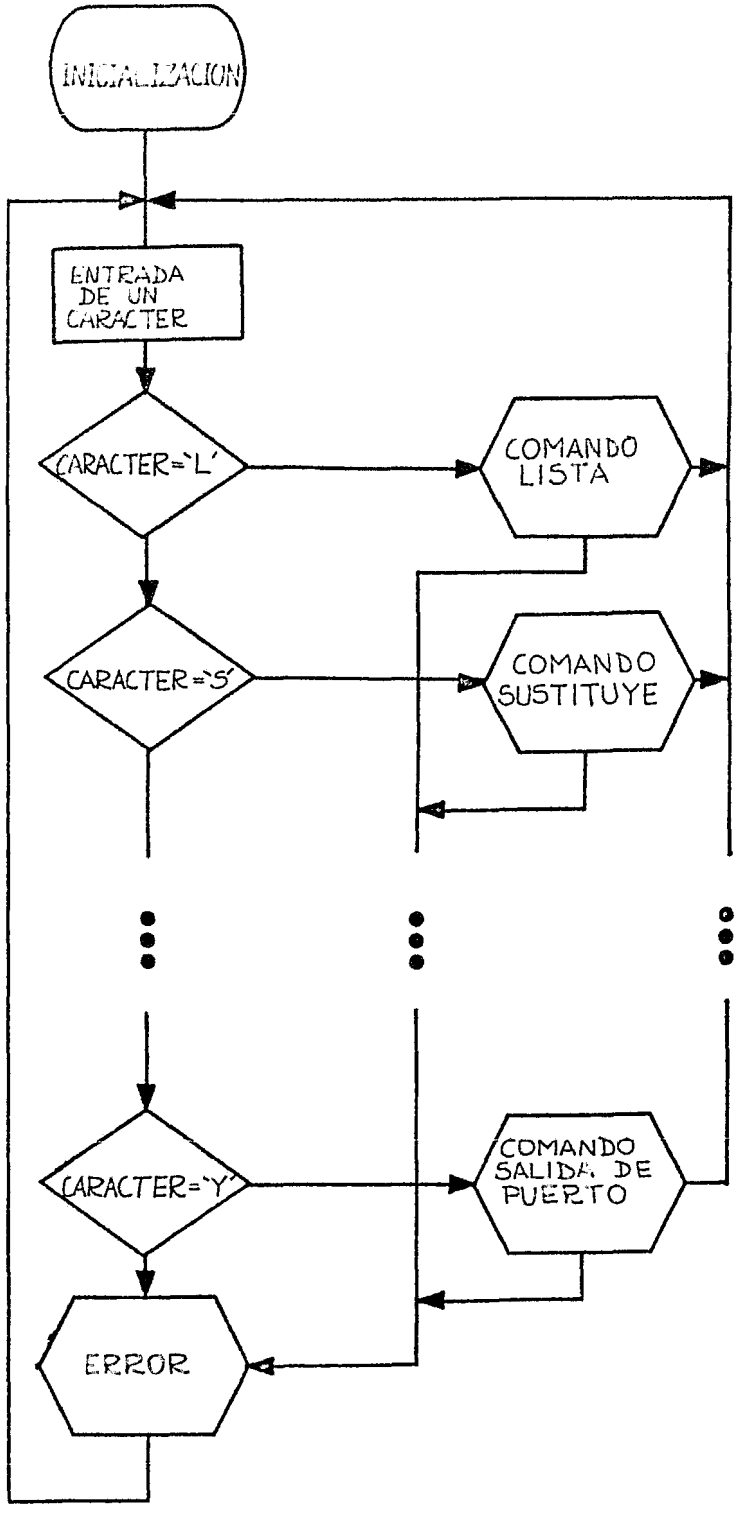


DIAGRAMA DE FLUJO DEL PROGRAMA MONITOR

siguiente comando.

Antes de entrar en detalle en el programa, veremos la forma de operación de los dos periféricos utilizados para el desarrollo del programa.

CTC (circuito temporizador y contador)

Descripción

El CTC es un circuito integrado que posee cuatro canales que pueden ser programados independientemente para operar como contadores utilizando una base de tiempo externa (un disparo externo) o tambien como temporizadores utilizando como base de tiempo un submùltiplo del reloj del sistema. Ademàs de proveer temporizaciòn y dividir frecuencias externas, el CTC puede servir como controlador de interrupciones para trabajar en el modo de interrupciòn 2 del Z80 con periféricos que no pertenezcan a la familia del mP Z80.

Cada canal del CTC es programado con dos bytes que deben ser enviados consecutivamente al canal programado en cuestiòn, debido a esto es recomendable que sean deshabilitadas las interrupciones cuando se programe un canal del CTC. En caso de que un canal del CTC sea habilitado para interrumpir al mP se deberà enviar el byte menos significativo del vector de interrupciones al canal 0 del CTC, este generarà los byte menos sig. de los vectores

correspondientes para los otros canales(ésto lo logra sumandole dos por cada vector al vector del canal cero).

Los cuatro canales del CTC deberàn ser programados con un primer byte o palabra de control donde se especifica la forma de operaciòn del canal y un segundo byte que se denomina constante de tiempo que representa el valor numerico por el que se va dividir la frecuencia interna o externa.

Estructura de un Canal del CTC

Un canal del CTC està compuesto de dos registros, dos contadores y lògica de control. Los registros son: registro de constante de tiempo de un byte y registro de control de un byte. Los contadores son: contador descendente de un byte que puede ser leido por el mP y un 'preescalador' de un byte. El registro de control de cada canal es escrito por el mP para seleccionar el modo y paràmetros de operaciòn de cada canal. El CTC posee cuatro registros de control correspondientes a los cuatro canales que posee el circuito. El canal que va a ser escrito o leido se determina por las lineas CS1 y CS2 del CTC las cuales se conectan por lo general a las lineas de direcciones A0 y A1 del mP.

Selección de un canal del CTC:

	CS1	CS2
Canal 0	0	0
Canal 1	0	1
Canal 2	1	0
Canal 3	1	1

El bit cero de la palabra de control deberà ser siempre 1 para que el CTC identifique esta palabra como palabra de control.

Modos de Operaciòn del CTC

Cuando el sistema es encendido el estado del sistema es indefinido. Cuando la señal de reset es activada el CTC es puesto en un estado conocido pero antes de que cualquier canal empiece su operaciòn deberà ser programado mediante la palabra de control y una constante de tiempo. Si alguno de los canales del CTC es habilitado para interrumpir al mP, una palabra deberà ser escrita en el canal cero la cual serà el vector de interrupciones para el canal 0 del CTC para los otros canales se generarà automàticamente. El CTC como se mencionó anteriormente puede ser programado como contador o como temporizador.

Modo de Contador del CTC

En este modo el canal del CTC se decrementa con los flancos en la entrada Clk/Trg del canal correspondiente . Para programar un canal del CTC como contador se coloca un 1 logico en el bit 6 de la palabra de control escrita en el canal. La entrada de reloj y disparo externo(Clk/Trg) es monitoreada durante una serie de flancos de disparo; luego de cada uno, en sincronia con el proximo flanco positivo del reloj del sistema Ø , se decrementarà el contador del canal. En cualquiera de los canales 0,1 y 2 cuando el conteo llega a cero aparece un pulso activo alto en la salida de "conteo cero"(ZC/TO) del canal correspondiente (por limitaciones del encapsulado del circuito, el canal 3 no posee esta salida). Cuando el contador llega a cero la constante de tiempo es cargada automàticamente en el contador y continua la operaciòn del canal. Si se escribe una cte. de tiempo nueva mientras el canal esta en operaciòn, el conteo serà completado antes de que se cargue la nueva cte. de tiempo. En este modo de operaciòn se pueden habilitar interrupciones para generar interrupciones desde otro periférico. Para lo anterior al generar la interrupciòn un periferico disparará el contador del canal programado y se iniciarà la cuenta que al llegar a cero interrumpirà al mP, de ésta se puede crear un retraso de la interrupciòn original del periferico y la

interrupciòn que llega al mP.

Modo de Temporizador

En este modo de operaciòn el canal del CTC trabaja como divisor del reloj del sistema Ø. Puede ser utilizado para tener bases de tiempo basadas en el reloj del sistema. En este modo el reloj del sitema es dividido por dos contadores en serie: primero el "preescalador" que divide el reloj del sistema por 16 o por 256, la salida del peescalador es utilizado como reloj para el contador descendente del sistema. Al igual que el modo de contador la cte. de tiempo es cargada automàticamente cada vez que el contador llega a cero y continua la operaciòn del canal.Ademàs al llegar a cero se produce un pulso en la salida ZC/TO dando como resultado un tren de pulsos de periodo preciso, dado por el producto: Ø * P * CT

Ø- Periodo del reloj del sistema

P- Factor de preescala, 16 o 256

CT- Constante de tiempo

El bit 3 de la palabra de control indicarà al canal si la operaciòn empezarà luego de cargar la cte. de tiempo o con un disparo externo en la entrada (CLK/TRG). Si la operaciòn empieza con un disparo externo, empezarà luego del segundo flanco positivo del reloj del sistema a partir del disparo. Hay que recordar que la señal de disparo deberà cumplir con

ciertas condiciones de tiempo requeridas por el circuito ya que de no ser asi podria ocurrir un disparo de muy corta duraciòn que no pueda ser detectado por el sistema.

Lògica de Interrupciones del CTC

La lògica de interrupciones del CTC asegura que éste actùe de acuerdo a los protocolos del sitema basado en un mP Z80.Para un sistema de este tipo las prioridades de interrupciòn estàn dadas por la localizaciòn fisica del dispositivo dentro de la configuraciòn "daisy chain" del sistema; en esta configuraciòn tiene mayor prioridad, el dispositivo mas cercano al mP, en el CTC la prioridad esta determinada por el numero de canal, teniendo la maxima prioridad el canal 0 y la minima el canal 3. De acuerdo al protocolo del sistema de interrupciones del Z80, los dispositivos y canales (del CTC) de menor prioridad no seràn atendidos al solicitar una interrupciòn si se està dando servicio a un dispositivo o canal de mayor prioridad. Al ser concluidas las rutinas de atenciòn de los dispositivos de mayor prioridad se atenderàn las solicitudes de los dispositivos o canales de menor prioridad. En el caso contrario un dispositivo de mayor prioridad, si interrumpirà una rutina de servicio de interrupciòn de un dispositivo de menor prioridad, la cual permanecerà pendiente hasta que termine la rutina del de mayor prioridad.

Cualquier canal del CTC y en cualquier modo de operaciòn puede ser programado para que interrumpa cuando su cuenta llegue a cero, en este caso el mP deberà ser programado en el modo 2 de interrupciones, esto con el objeto de que cuando ocurra la interrupciòn y sea reconocida por el mP el CTC envie al MP el vector de interrupciones correspondiente al canal que interrumpiò.

Programaciòn de la Palabra de Control

Cada uno de los ocho bits de la palabra de control, para cada uno de los canales del CTC, deberà ser programado para establecer el modo de operaciòn del canal correspondiente del CTC.

Palabra de Control:

D7 interrupciones

1 habilita interrupciones

0 interrupciones no habilitadas

D6 Modo de Operaciòn

1 Modo de Contador

0 Modo de Temporizador

D5 Factor de Preescala (solo para modo de Temp.)

1 Valor de Preescala=256

0 Valor de Preescala=16

D4 Selecciòn del Flanco de disparo (entrada CLK/Trg)

1 Flanco positivo

0 Flanco negativo

D3 Forma de disparo(solo para modo de Tem.)

1 Pulso de disparo en CLK/Trg empieza operación.

0 Disparo automático: al cargar la cte. de tiempo del canal, empieza operación.

D2 Constante de Tiempo

1 Prosigue palabra de cte. de tiempo

0 No prosigue palabra de cte. de tiempo

1 Reset del canal por software

0 Operación continua del canal

D0 Control o Vector

1 Palabra de control

0 Vector de Interrupciones

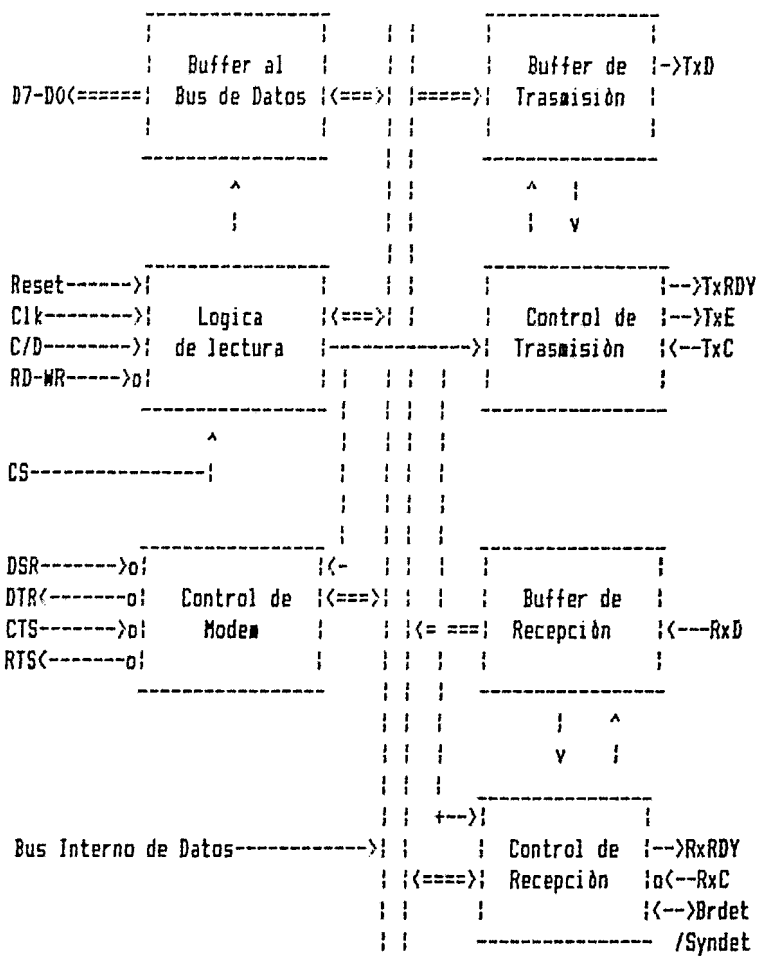
USART (8251A INTEL)

El USART 8251A de Intel es un puerto serie programabl diseñado para comunicaciòn del microprocesador 8085 de Intel y puede ser programado para operar en forma asincrona o en forma sincrona. El USART acepta datos en paralelo a través del bus de datos del sistema y los convierte a una corriente de datos en serie, simultaneamente puede recibir datos en serie y convertirlos en datos en paralelo para el CPU. El USART tiene señales para indicar al CPU cuando esta listo para enviar un dato o si aceptò un dato que llego al sistema en serie.

Señales de Control del Usart:Reset, Clk, WR, RD, C/D Y CS; mediante estas señales se determina la operaciòn que realiza el USART en un momento determinado.

Señales para Modem:estas señales para dispositivos moduladores, demoduladores son de tipo "Handshake" y permiten mantener una conversaciòn de control del dipositivo Modem con el Usart.Estas señales son: DSR, DTR, CTS Y RTS.

Diagrama de bloques del Usart 8251A de Intel.



TRASMISION DEL USART

Buffer de trasmisiòn: este buffer acepta datos en paralelo del bus de datos del sistema y los convierte a serie, ademàs de insertarle al dato los bits utilizados en las tecnicas de trasmisiòn serie. La cadena de bits en serie saldrà a través de la pata TxD del USART.Ademàs la trasmisiòn deberà estar

habilitada por la pata CTS=0.

Control de Trasmisiòn

TxRDY (Trasmisiòn Lista): Esta señal indica al CPU que el USART esta listo para trasmitir un dato. La pata de salida TxRDY puede ser empleada en un sistema que utilice Interrupciones (esta salida puede ser mascarada por el bit TxDisable al programar el 8251A) o la señal puede ser leida del status(bit TxRDY) para un sistema mediante "Polling". Cada vez que se escribe un dato al usart esta señal serà desactivada.

TxE (Trasmisor Vacio): Cuando el 8251A no tiene caracter para trasmisiòn se activa esta señal que indica que el buffer de salida esta libre. Esta señal nos indica el fin de la trasmisiòn de un caracter y nos puede ser ùtil en el caso de trasmisiòn "Half Duplex" en este caso nos permite cambiar la linea de trasmisiòn a entrada. Para que esta señal no pueda confundirse con la anterior, hay que aclarar que esta indica que el buffer de trasmisiòn esta vacio mientras que la anterior solo indica que el buffer que nos conecta al bus de datos del sistema, esta vacio.

TxC (Reloj de trasmisiòn): La señal de reloj que se aplica a esta entrada controla la velocidad de trasmisiòn de los

caracteres. En el modo sincrono el baudaje sera igual a la frecuencia de la señal en TxC, y en el modo asincrono; el baudaje sera una fracciòn de esta frecuencia. La fracciòn se determina cuando se programa el 8251A y puede tener los siguientes valores: 1, 1/16 y 1/64. Como ejemplo tendriamos que; para un baudaje de 300baudios se necesitarian las siguientes frecuencias 300Hz(x1), 4800Hz(x16) o 19.2KHz(x64). Esto para las diferentes opciones de programaciòn. Con el flanco negativo de la señal TxC se va corriendo el contenido del buffer de salida hacia la pata TxD.

RECEPCION DEL USART

Buffer de Entrada: Acepta datos en forma serie y los convierte a formato paralelo, revisa los bits o caracteres especiales de comunicaciòn y mantiene un caracter "Ensamblado" listo para que sea leido por el CPU. El dato en formato serie entra a través de la pata RxD.

Control de Recepciòn

RxRDY (Recepciòn lista): Esta señal indica que el 8251A contiene un caracter ensamblado que puede ser leido por el CPU. La pata RxRDY puede ser conectada al sistema de Interrupciones y de esta forma utilizar Polling el bit RxRDY puede ser leido de status. La señal en la pata de salida

puede ser mascarada por RxEnable mas no el bit en el status.

RxC (Reloj de Recepciòn): Este es similar al de trasmisiòn y en la mayoria de las aplicaciones se unen estas entradas. La unica diferencia estriba en que el dato que entra al 8251A se va recorriendo con el flanco positivo de la señal RxC.

Break Detect (Detecciòn del caracter de ruptura): esta señal es solo para el modo Asincrono y es activada cuando se detecta una palabra en que todos los valores son cero (incluyendo bit de inicio, bits de datos, bit de paridad y bits de parada). Esta señal solo puede ser desactivada con la señal de Reset.

OPERACION DEL USART 8251A DE INTEL

Las funciones que realice el 8251A estaràn determinadas por la forma como sea programado éste. El CPU deberà enviar un grupo de palabras de control para inicializar el Usart, esto, de acuerdo al formato de trasmisiòn que se utilizara. Mediante las palabras de control se puede programar: baudaje, longitud de caracteres, número de bits de parada, modo asincrono o sincrono, tipo de paridad(par o impar). En el modo sincrono se puede seleccionar entre un caracter de

sincronia interno o externo.

Una vez que el 8251 sea programado, estarà listo para realizar las funciones de comunicaciòn serie requeridas por el sistema. La salida TxRDY se pondrà activa para indicar al CPU que esta listo para recibir un dato de este y enviarlo al exterior del sistema en forma serie. Esta salida es desactivada una vez que el CPU coloca un dato en el Usart. Simultàneamente el Usart puede estar recibiendo un dato en forma serie de un Modem u otro dispositivo, al terminar la recepciòn del caracter activarà la señal RxRDY para señalarle al CPU que puede leer un dato del Usart. Esta señal es desactivada una vez que el dato ha sido leido.

El 8251 no podrà empezar a trasmitir hasta que el bit TxEnable de la instrucciòn de comando sea puesto en 1.

Programaciòn del 8251A

Antes de empezar la trasmisiòn o recepciòn de datos el Usart deberà ser programado por un grupo de palabras de control enviadas por el CPU. Estas palabras de control son:

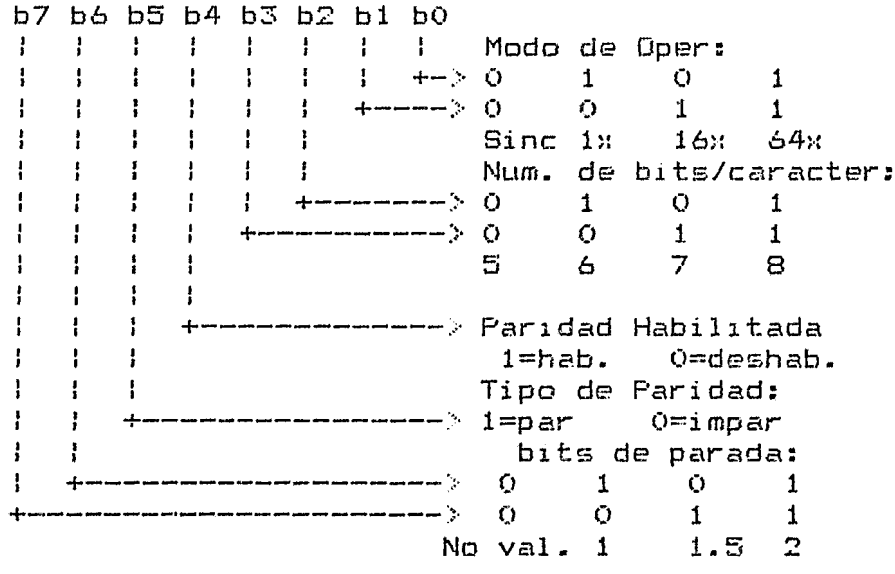
1.-Instrucciòn de Modo.

2.-Instrucciòn de Comando.

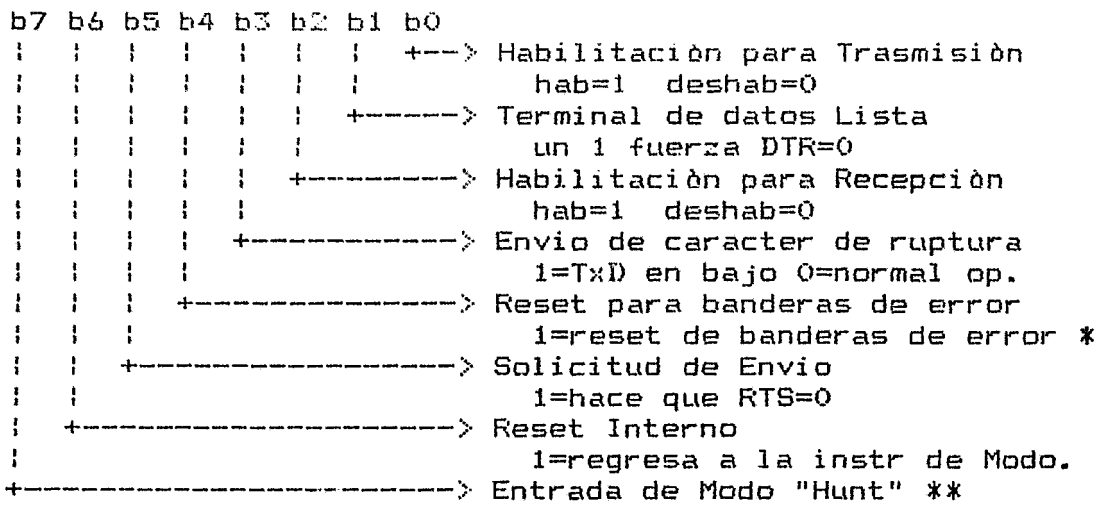
Instrucciòn de Modo

Esta palabra indicarà la operaciòn general del 8251, deberà enviarla el CPU luego de un Reset (por software o hardware). Una vez escrita esta palabra deberà escribir en el Usart el caracter de sincronìa (para modo sincrono) o la Instrucciòn de comando por parte del CPU.

Palabra de Instrucciòn de Modo



Palabra de instrucciòn de Comando



* Las Banderas de error no afectan la operaciòn del Usart, solo sirven para informar si ocurriò un error en el Usart:
1.-Paridad: se enciende si hay un error en la paridad del caracter recibido.
2.-"Overrun": se enciende si el CPU no lee un caracter del Usart y el siguiente llega al Usart (traslapan).
3.-Formato: se enciende solo en el modo asincrono e indica, que no se detecto ningùn bit de parada.

** Este Modo no afecta cuando operamos en forma Asincrona.

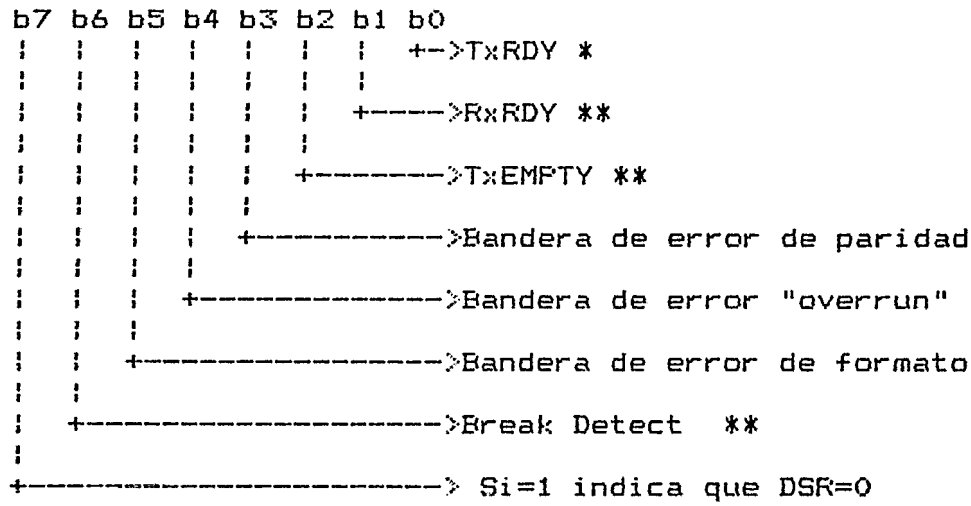
Definiciòn del Status del USART 8251A

En comunicaciòn de datos muy a menudo es necesario revisar el "status" para ver si ha ocurrido algùn error o ver algunas condiciones que requieran especial atenciòn por parte del mP. El status del 8251A puede ser leido en cualquier momento mientras està en operaciòn (la actualizaciòn del status se inhibe mientras se esta leyendo dste. Para leer el status el mP pondra la entrada C/D=1

además de activar la señal de lectura.

Algunos bits del status tienen idéntica función que algunas de las patas del usart. Esto con motivo de tener la opción para utilizar el Usart en un sistema mediante interrupciones o para un sistema por polling. La pata TxRDY es una excepción.

Palabra de Status



* TxRDY tiene diferente significado que la pata de salida.
TxRDY (status)=Buffer de dato vacio.
TxRDY (pata)=Buffer de dato vacio*(CTS=0)*(TxEN=1)
** Igual que las patas de salida.

Inicialización del Programa Monitor

Antes que todo, lo primero es inicializar el apuntador de Pila para poder utilizar las rutinas del programa sin

temor a que surjan errores. Luego inicializamos el canal cero del CTC para que genere la señal de reloj para el baudaje. Ahora bien utilizaremos el modo de temporizador del CTC dividiendo la señal de reloj del sistema entre 16. Con esto tendremos una señal base de una frecuencia:

Fo= ∅/16 Hertz

∅= Frecuencia del pulso del Reloj del Sistema

pero debido a una restricciòn del Usart la señal tuvo que ser dividida con un flip-flop entre dos por lo que la señal base es de:

Fo'= ∅/32 Hertz

a partir de esta frecuencia vamos a generar los diversos baudajes que se necesiten. Para nuestro equipo la Frecuencia del pulso del Reloj del sistema es de 2,000,000 Hertz y ya que el canal del CTC divide la señal por el factor cte de tiempo tendremos que el baudaje serà igual a:

Fb=∅/(32*CT).

∅=Frecuencia del pulso del Reloj del sistema.

CT=Constante de Tiempo para el CTC.

de aquí podemos determinar las frecuencias que podemos generar ya que la constante de tiempo tiene un rango de 1..255.

Dentro de este rango solo podemos generar baudajes estandar ya que estaremos conectados a una terminal de [illegible] entremilesimientos bits por segundo [illegible]. Incrementando la fracción anterior CT y

sustituyendo por valores comerciales para la frecuencia de baudaje encontramos 6 diferentes valores de CT para los diferentes baudajes.

$$CT=2,000,000/(Fb*32)$$

de aqui se genero la siguiente tabla:

Tabla de Constantes de tiempo

Baudaje	CT dec	CT hex
300	208	D0
600	104	68
1200	52	34
2400	26	1A
4800	13	0D

Con los valores que se dan en la tabla hay una desviaciòn del valor exacto de baudaje por lo que se cometen errores fraccionarios en la fijaciòn del baudaje. Para la selecciòn del baudaje se lee el puerto "BAUDR" el cual no es mas que un buffer conectado al bus de datos para poder leer un grupo de interruptores los cuales estàn asociados, cada uno a los baudajes que podemos generar. Como los interruptores estàn conectados a resistencias de "pull-up" por un lado y por el otro a tierra, cuando estén cerrados darán a la salida un "0" lógico y abiertos un "1" de esta manera podemos leer un numero dentro del cl. Como se puede

observar en la tabla los demas valores son submultiplos de dos a partir de 208, esto nos permite escojer nuestra constante simplemente dividiendo 208 entre N*2 (N es igual a el número leido en el puerto BAUDR).

Selección de la cte. de tiempo para un baudaje dado.

```
Baudsel  IN      A,(BAUDR)   ;lee valor escogido
                             ;de baudaje
         LD      C,0D0H      ;C=constante para 300 bauds
CORRE    SRA     A           ;corre ac. hacia la
                             ;bandera de Cy.
         JR      NC,SAL      ;si el bit=0 termina
         SRL     C           ;si el bit(n)<>0
                             ;entonces C=C/2
         JR      CORRE       ;verifica siguiente bit
SAL      LD      A,C         ;Carga el Ac. con la cte.
                             ; de tiempo.
```

Una vez seleccionada la constante de tiempo para el CTC se programa éste enviando las dos palabras de control. Ya que se realizò esto, se programa el Usart en modo Asincrono ya que es el formato de operaciòn de la mayoria de las terminales comerciales. Para empezar a programar el Usart debemos darle un Reset por software para asegurar su correcta programaciòn sin tener que "resetear" todo el sistema. Para esto enviamos un caracter cualquiera al Usart para asegurar que estemos en la instrucciòn de comando y desde aqui podemos darle el reset enviando un 04 al Usart desde el mP. Un vez que el Usart ha sido "reseteado" podemos programarlo sin temor a equivocaciones y solo restarà enviar la palabra de instrucciòn de modo(forma de operaciòn) y la palabra de instrucciòn de comando (habilitaciòn de algunos paràmetros). La palabra de instrucciòn de modo permite en éste caso el [illegible], factor de baudaje :1, 8

caracteres de datos, sin bit de paridad y un bit de parada. Lo anterior da como resultado un 4Dhex. Para el caso de la instrucciòn de comando, habilitamos las salidas TxRDY y RxRDY, ademàs de no "resetear" el usart en esta ocasiòn; los otros paràmetros se deshabilitaron. Esto dió como resultado un 25hex. Ya que tenemos todos los datos, hacemos una tabla con éstos y los enviamos al usart para programar éste. Para realizar lo anterior se utiliza la instrucciòn compleja del mP Z80 OTIR la cual envia una tabla de datos a un puerto, a la cual apunta el par de reg. HL, el reg. B contiene el nùmero de datos de la tabla (4 en éste caso) y C el valor del puerto donde se enviarà (usart+1).

Una vez inicializado el CTC y el Usart podemos comunicarnos con la terminal. Para que el usuario tome conocimiento de esto se pinta en la terminal un letrero en el caso particular de la terminal Televideo modelo 910 que se utilizò, se envió antes del letrero la secuencia de caracteres ASCII: escape y '*' con lo cual se borra la pantalla de la terminal. Despues de esto pasamos al programa en si con el reconocimiento de comandos. Este se hace de una manera comun, leyendo un caracter de la terminal y viendo si éste està en un rango entre los caracteres 'A' y 'Z'. SI està dentro de éste rango deberà ir a un comando y si no irà la rutina de error. Para algunas letras dentro del rango dado, no corresponde ningun comando por lo que el control del programa también se transfiere a la rutina de error.

Rutinas del Programa Monitor.

Las rutinas del programa monitor fueron la parte màs importante en el desarrollo de programa.Las rutinas que utiliza el programa monitor las podemos dividir en tres grupos segùn el uso dado a éstas: basicas, generales y particulares. El primer grupo lo forman aquellas rutinas necesarias para la comunicaciòn con el sistema, el segùndo las rutinas màs utilizadas en todo el programa y el ùltimo lo forman rutinas especìficas de ciertos comandos solamente.

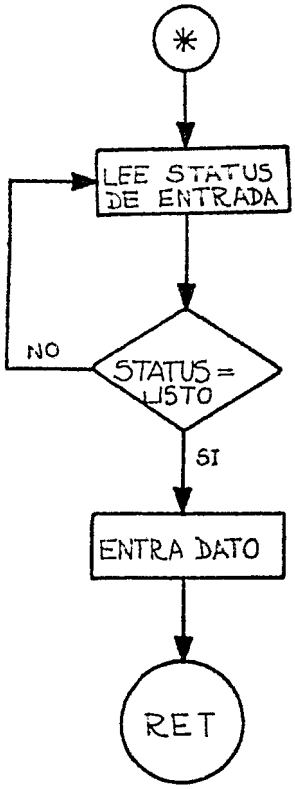
Rutinas Bàsicas.

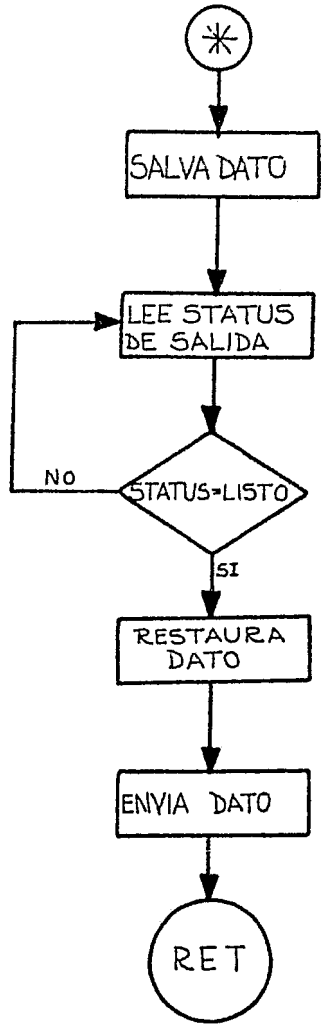
Aquì también podemos dividir entre dos grupos bien definidos: rutinas de entrada-salida y rutinas de conversiòn de codigo.

Las rutinas de Entrada y Salida que tenemos son:

POLENT:esta rutina es la bàsica de entrada. Lee el status de entrada del Usart y espera a que haya un dato vàlido en el buffer de entrada. Al haber un dato regresa con éste al programa.

SALTTY:esta es la rutina bàsica de salida,cuando se le llama el àcumulador deberà tener el caracter ASCII





RUTINAS BASICAS DE ENTRADA Y SALIDA

POLENT
(Rutina basica de entrada)

SALTTY
(Rutina basica de salida)
En ésta rutina se salva el dato, ya que al leer el satatus del Usart se destruye el contenido del Ac.

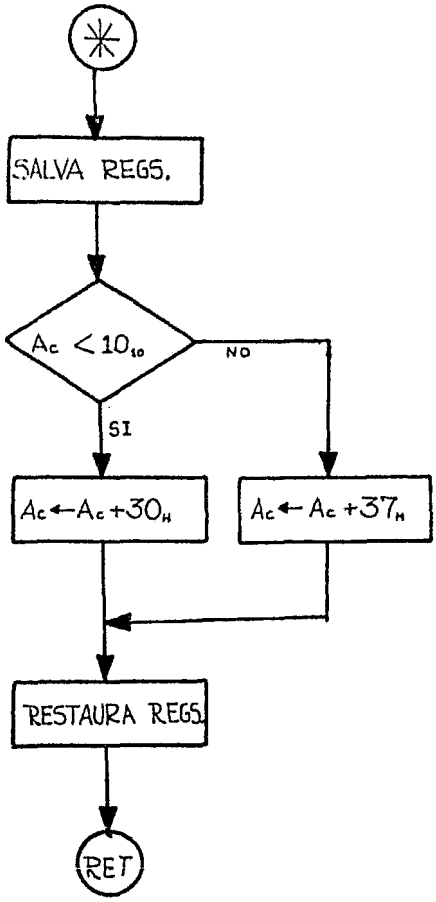
que se [illegible] a la terminal. Este es salvado mientras se lee el status de salida del Usart y el mP espera a que el Usart esté listo para enviar el dato. En este momento se restaura el caracter a ser enviado y se envia. Después de esto se regresa a donde se hizo la llamada.

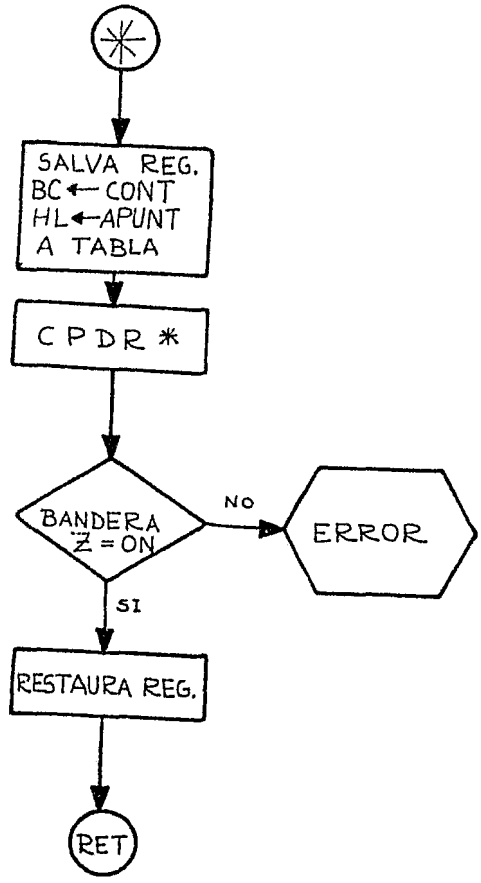
Debido a que la trasmisiòn se realiza mediante el uso de caracteres ASCII e internamente se utiliza còdigo binario para las operaciones es necesario contar con rutinas de conversiòn para los dos còdigos.

Las rutinas bàsicas de conversiòn que tenemos son:

CONV: esta rutina nos convierte del codigo ASCII a hexadecimal. Para esto se pasa el caracter ASCII en el àcumulador y el equivalente hex. regresa en el reg. C . Para esta rutina se utiliza una tabla y una instrucciòn de comparaciòn del mP Z80 (CPDR), aunque no es totalmente eficiente porque aumenta el numero de operaciones para realizar la conversiòn. Se diseñò asi la rutina para apreciar éste tipo de instruccion.

HEX: esta rutina realiza la funciòn inversa de la anterior. Convierte los cuatro bits menos [illegible] del acumulador en el caracter ASCII





RUTINAS BASICAS DE CONVERSION

HEXCO
(Rutina de Hex. a ASCII)

CONV
(Rutina de ASCII a Hex.)
*La instrucción CPDR compara el Ac. con una tabla y cuando un valor de la tabla=Ac la badera Z="ON".

correspondiente. El caracter NULL retorna también en el àcumulador.

Rutinas Generales.

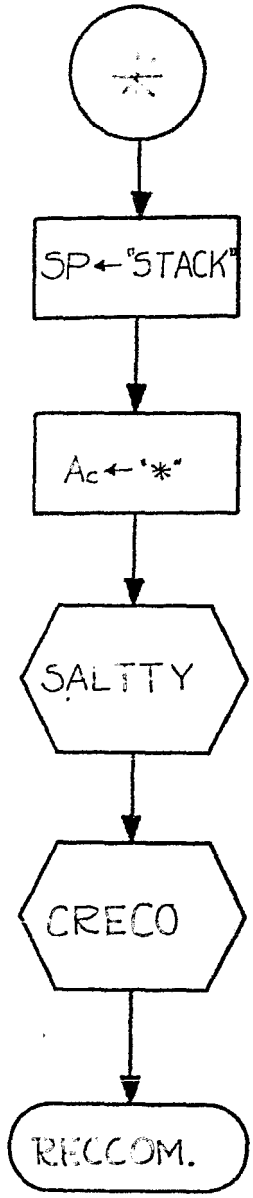
En el caso de rutinas generales la mayoria son de entrada y salida, por lo que no se haràn divisiones en éste grupo.

En éste grupo tenemos las sguientes rutinas:

ERROR: es la màs pequeño de todos, pero la màs utilizada en el programa, ya que es llamada cada vez que ocurre un error dentro de una rutina, comando y cuando se estàn reconociendo los comandos. Esta rutina reinicializa el apuntador de pila ya que si el error ocurriò dentro de una rutina el apuntador de pila ha sido afectado. Por otra parte envìa un asterisco a la terminal como señal de que ocurriò un error y llama a la rutina CRECO para después de esto terminar (regresar al reconocimiento de otro comando).

CTRLC: ésta es una rutina de entrada que sirve para abortar un comando cuando se està ejecutando. Para esto el mP lee el status de entrada del Uart y si no tiene ningun dato en su buffer de entrada

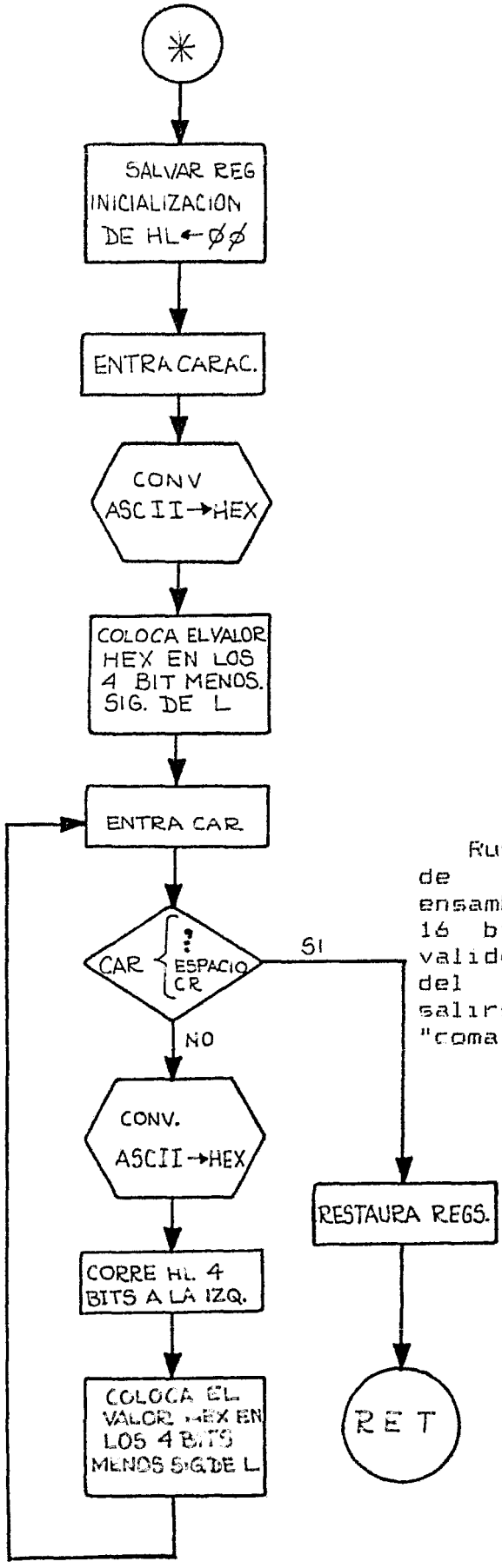
RUTINA DE ERROR



regresa sin causar ningun cambio al comando. En caso de llegar un dato verifica si es el caracter "escape" o "ctrl c", los cuales hacen que se envie el control del programa a la rutina de error con lo que se aborta el comando; por otro lado si entra un caracter diferente a éstos la rutina lo desecha y regresa sin efectuar cambio alguno al comando.

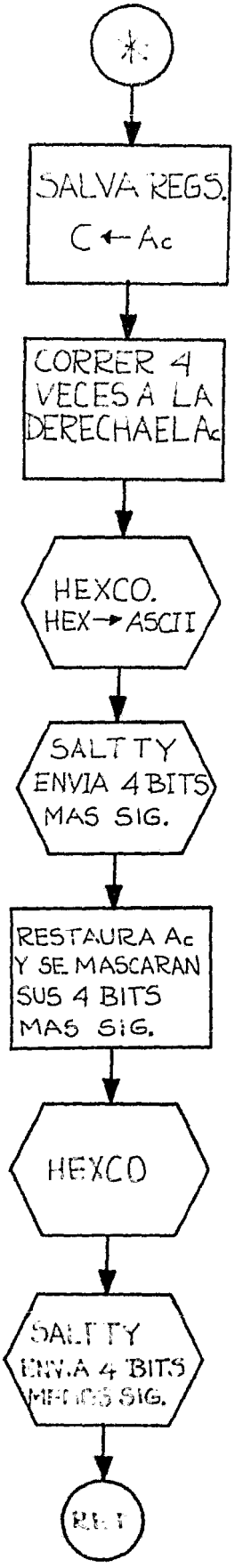
CO16: ésta es una de las rutinas màs utilizadas en el programa y por la mayoria de los comandos. Permite la entrada de hasta cuatro caracteres ASCII equivalentes a hex. y los convierte a un nùmero de 16 bits en los registros HL del mP Z80. Para realizar esto llama a POLENT y a CONV. Para poder enmendar errores esta rutina permite que se tecleen los nùmeros hex. indefinidamente y solo cuando recibe un caracter de control acepta los ultimos cuatro digitos hex. que fueron teclados. Si se teclea un caracter que no es vàlido se va a error y se aborta el comando.

BYTTY: esta rutina permite desplegar el contenido del àcumulador en hexadecimal, mediante el envio de dos caracteres ASCII a la terminal. Esta rutina llama a HEXCO y a SALTTY principalmente.

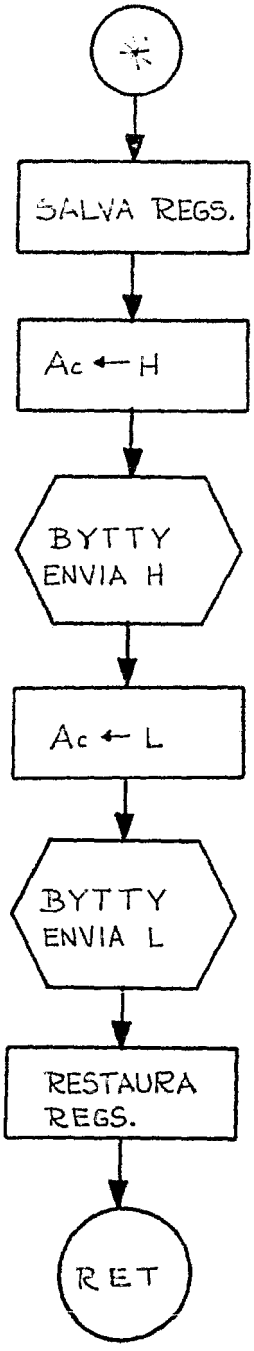


C016

Rutina de uso general para la entrada de cuatro caracteres ASCII que se ensamblan como un valor Hexadecimal de 16 bits en el par de reg. HL (seràn validos los ùltimos cuatro digitos antes del caracter de control que permite salirse de la rutina: "espacio", "CR", "coma" o ":").


SALVA REGS.
C ← Ac
CORRER 4 VECES A LA DERECHA EL Ac
HEXCO.
HEX → ASCII
SALTTY
ENVIA 4 BITS MAS SIG.
RESTAURA Ac Y SE MASCARAN SUS 4 BITS MAS SIG.
HEXCO
SALTTY
ENVIA 4 BITS MENOS SIG.
RET

BYTTY


SALVA REGS.
Ac ← H
BYTTY
ENVIA H
Ac ← L
BYTTY
ENVIA L
RESTAURA REGS.
RET

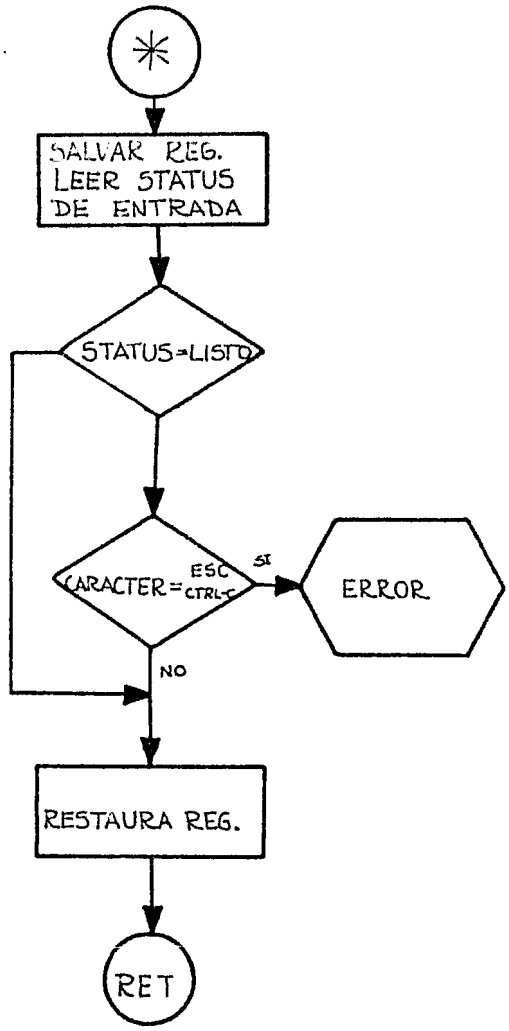
SALHL

RESTA: esta rutina hace la resta de 16 bits y podríamos decir que su función es HL=HL-DE+1. Esta rutina se utiliza para el cálculo del rango de memoria que será afectado por algún comando. El incremento de la diferencia en uno se debe a que el comando debe afectar desde la primera locaiidad hasta la última que se pide al llamar al comando y que la diferencia es el contador para el comando.

SALHL: esta rutina nos permite desplegar el contenido de los reg. HL en hex. enviando cuatro caracteres ASCII a la terminal. Esta llama a BYTTY.

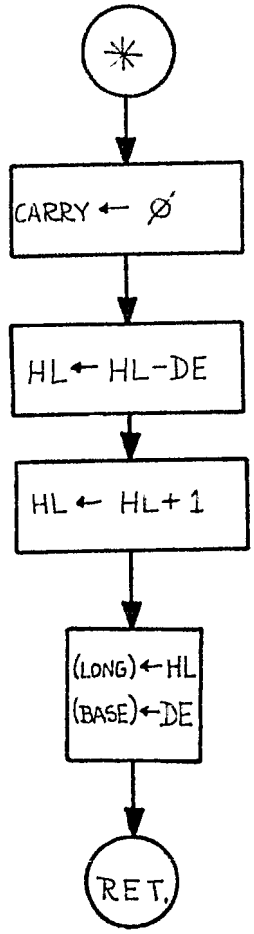
CRECO: esta rutina envia una secuencia de caracteres cuando deseamos saltar de linea. Envia un "retorno de carro", un "salto de linea" y un caracter "prompt". Además esta rutina es llamada cada vez que se recibe un caracter "retorno de carro (CR)".

ECO: mediante esta llamada podemos rebotar los caracteres que llegan a nuestro sistema de la terminal. Esta rutina determina si el caracter se rebotará tal y como llegó, no se rebotara (algunos caracteres pueden bloquear la terminal) o en el caso de un "CR" se llamara a CRECO.



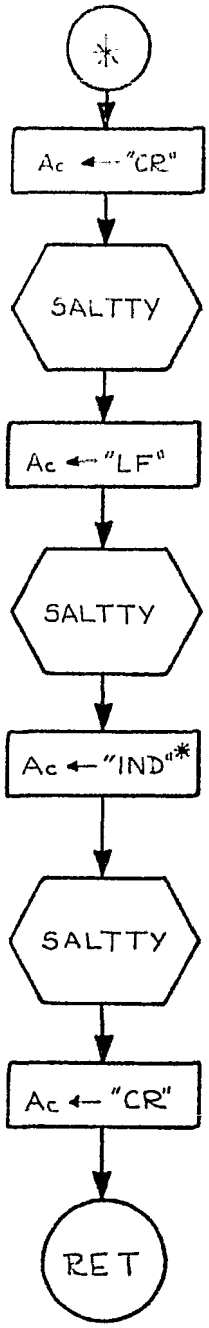
CTRL-C

Se utiliza para abortar los comandos durante su ejecución.



RESTA

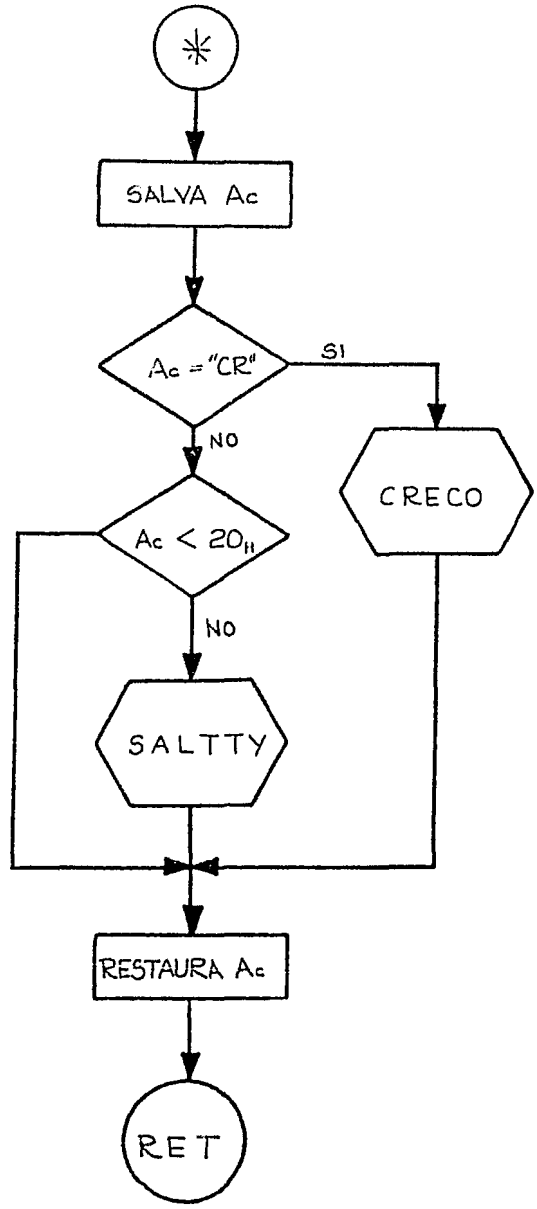
Calcula la diferencia de dos valores hex. de 16 bits e incrementa su resultado en uno.



CRECO

Rebota el caracter CR o salto de linea en el Programa Monitor.

*-IND: éste es un caracter que indica que estamos bajo el control del Programa Monitor.



ECO

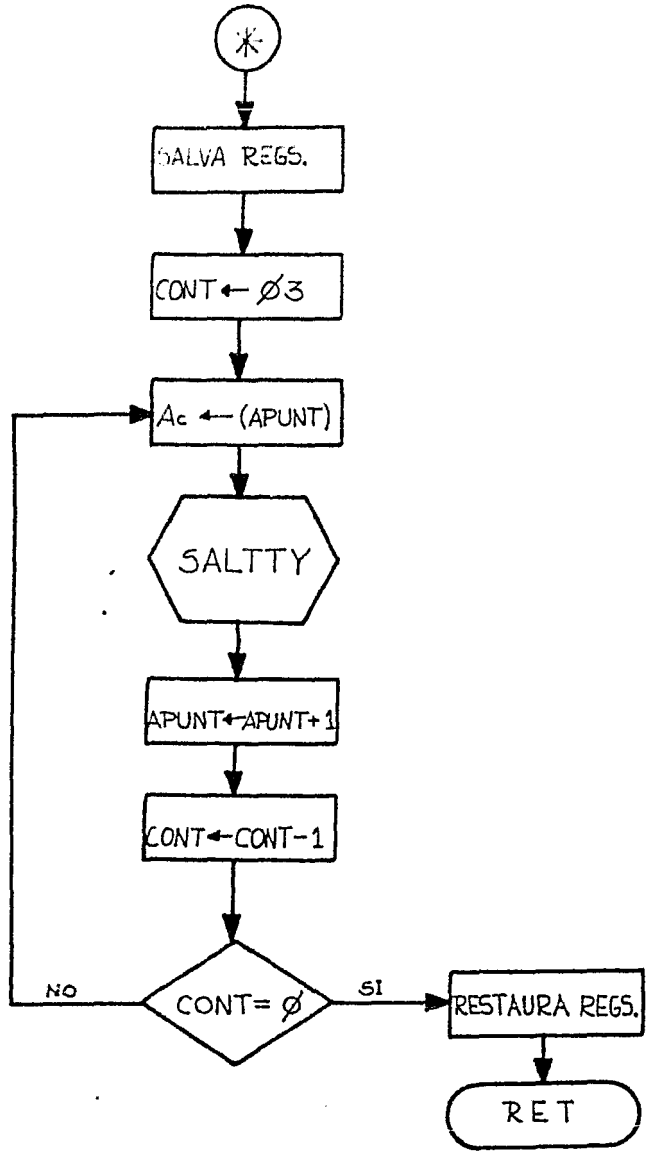
Rebota los caracteres que llegan al Starter kit desde la terminal hacia ésta.

Rutinas Especificas del Monitor.

PINTA: esta rutina despliega en la terminal tres caracteres ASCII de una tabla cada vez que es llamada, e incrementa un apuntador dentro de la tabla. Esta tabla contiene los caracteres correspondientes a los diversos registros del mP Z80, esta llamada solo la utiliza el comando para desplegar registros.

BAND: esta llamada despliega las banderas del mP con los caracteres ASCII correspondientes a las diversas banderas que tiene el mP Z80 (Z,C,H,P,V). Esto lo hace pasando de memoria al àcumulador el valor del reg. F que se guarda en una localidad determinada, después corre el àcumulador hacia la bandera de Carry y si està encendida envia un caracter correspondiente a la posiciòn del bit.

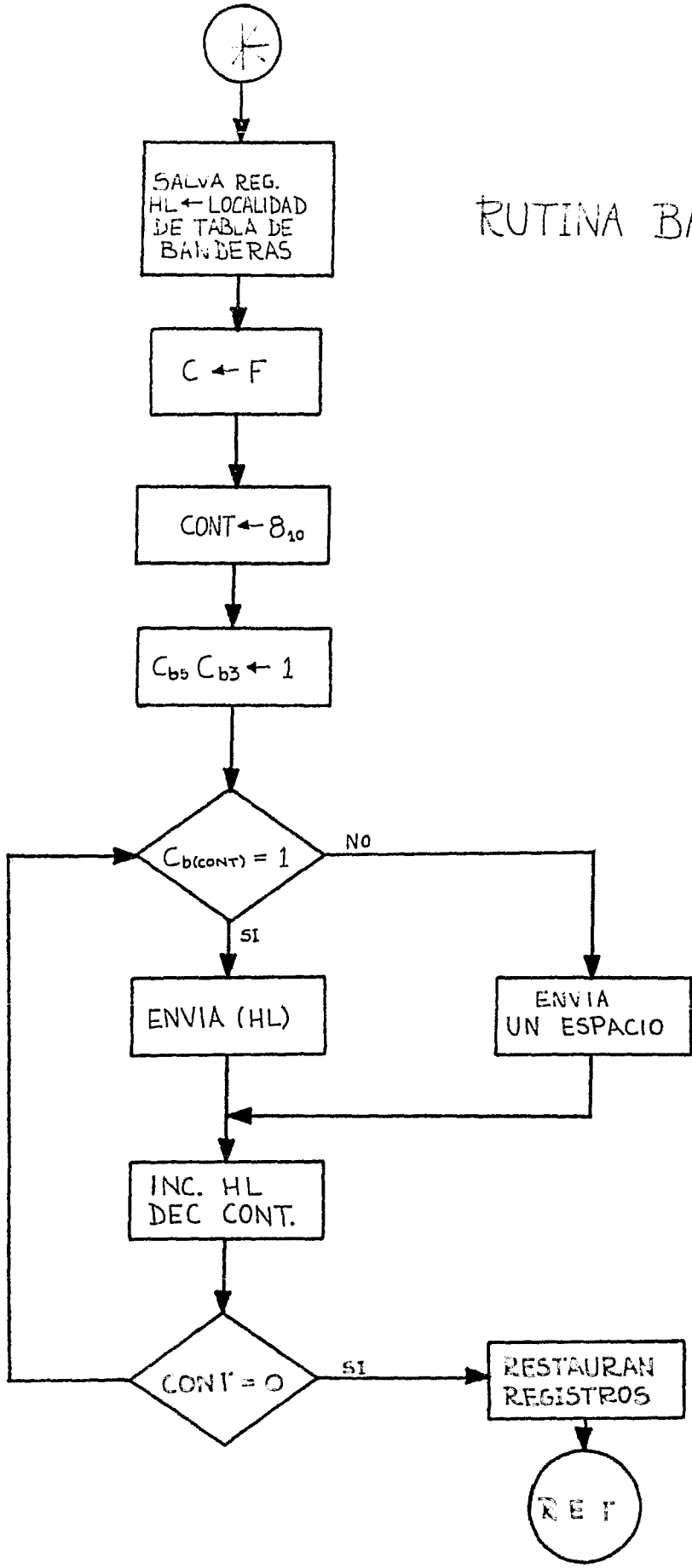
RST8:esta rutina puede ser considerada casi un comando ya que regresa siempre al reconocimiento de comandos. Se llama con un RST 08H y su funciòn es salvar todos los registros del mP en memoria Ram, quita los puntos de ruptura y retornar al monitor. Desde esta rutina se pueden examinar el

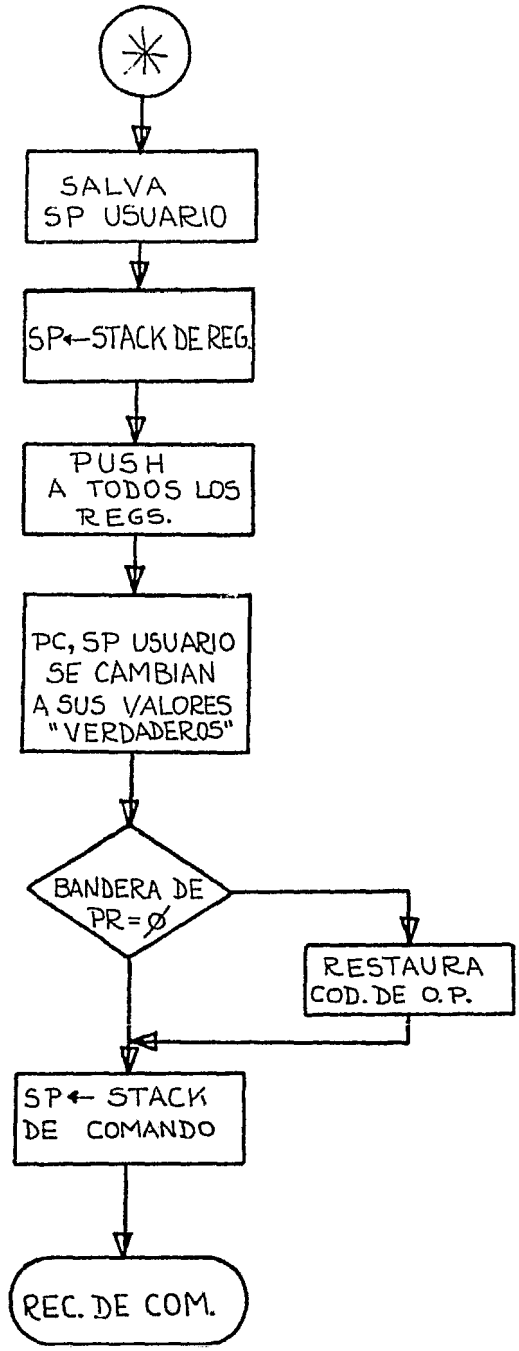


PINTA

Envia tres caracteres de una tabla que contiene los diversos caracteres correspondientes a losregistros internos del mP. El apuntador esinicializado antes de

RUTINA BAND

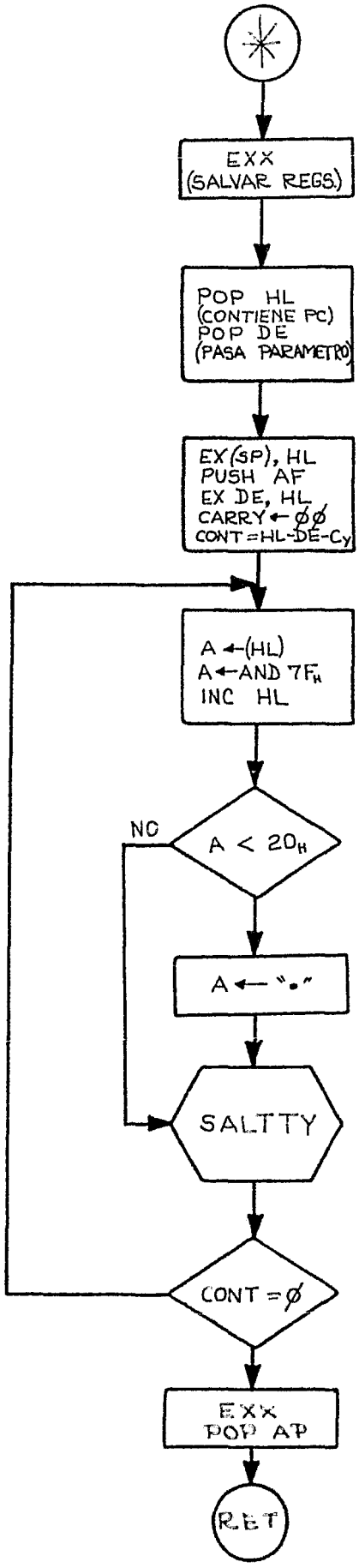

SALVA REG.
HL ← LOCALIDAD DE TABLA DE BANDERAS
C ← F
CONT ← 8₁₀
C_b5 C_b3 ← 1
C_b(CONT) = 1
NO
SI
ENVIA (HL)
ENVIA UN ESPACIO
INC. HL
DEC CONT.
CONT = 0
SI
RESTAURAN REGISTROS
RET


SALVA
SP USUARIO
SP←STACK DE REG.
PUSH
A TODOS LOS
REGS.
PC, SP USUARIO
SE CAMBIAN
A SUS VALORES
"VERDADEROS"
BANDERA DE
PR=Ø
RESTAURA
COD. DE O.P.
SP← STACK
DE COMANDO
REC. DE COM.

RUTINA RST 8.

contenido de los diversos regs. del mP Z80 y cambiarlos. Se le puede llamar automaticamente con los puntos de ruptura o con el codigo CFhex que corresponde a un RST 08H.

PASCII:esta rutina nos sirve para desplegar el contenido de memoria en valores ASCII y se llama dentro del comando Lista. Esta rutina recibe la localidad inicial y final que seran desplegadas a través de la pila, luego calcula el rango y va desplegando los contenidos de memoria que puedan ser desplegados en éstos. Si el contenido no puede ser desplegado por ser caracter de control envia un caracter "punto" a la terminal.



PASCII

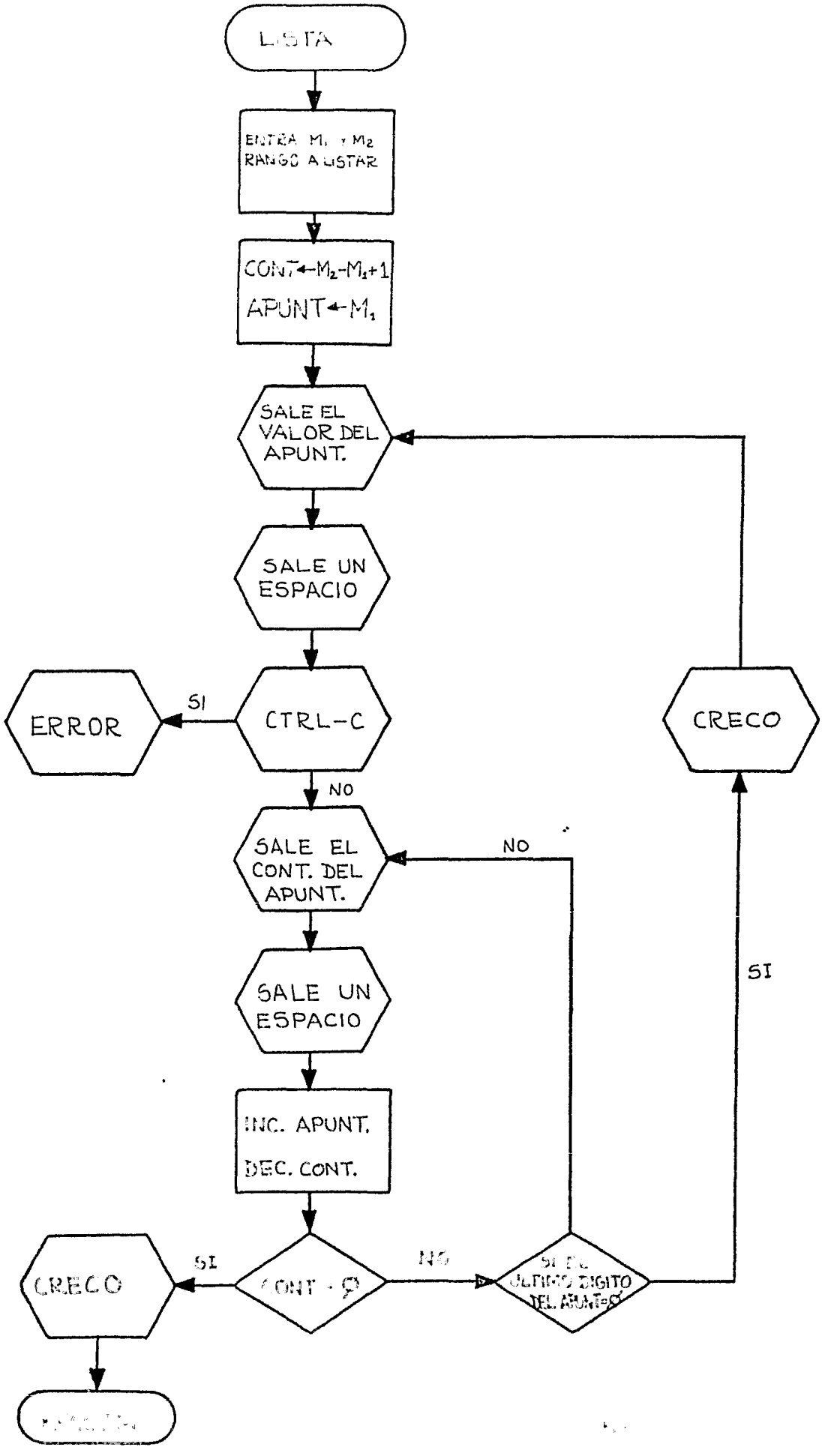
Despliega el contenido de un rango de memoria como caracteres ASCII. En caso de caracteres de control se envia un punto a la terminal.

Comandos del Programa Monitor para Terminal.

Una vez que se han descrito las diversas rutinas con que cuenta el programa podemos empezar a explicar el funcionamiento de los comandos del programa. Antes que todo hay que recordar el diagrama de flujo del programa para entender que todos los comandos regresan al mismo punto de donde partieron: el reconocimiento de comando.

Ahora haremos la descripcion de los diversos Comandos

COMANDO LISTA: éste comando es llamado con el caracter 'L' y se le pasa como paràmetro la localidad inicial y la localidad final que deseamos ver; con esta finalidad se llama a CO16 y a RESTA . Este comando despliega los contenidos de las localidades de memoria desde la primera localidad dada hasta la segunda. El contenido de una localidad va separado del siguiente por un espacio. Se despliegan en hexadecimal y en grupos de dieciseis valores hex. por linea. La ultima localidad de la linea sera la terminada en FH Dentro de este comado llamamos a CTRLC por lo que mientras dentro de esta rutina se está listando los contenidos de las localidades. El numero de localidades que podemos


LISTA
ENTRA M1 y M2 RANGO A LISTAR
CONT←M2-M1+1
APUNT←M1
SALE EL VALOR DEL APUNT.
SALE UN ESPACIO
CTRL-C
SI
ERROR
NO
SALE EL CONT. DEL APUNT.
NO
SALE UN ESPACIO
INC. APUNT.
DEC. CONT.
CONT = 0
SI
CRECO
NO
SI EL ULTIMO DIGITO DEL APUNT=8
SI
CRECO

observar simultáneamente depende del número de líneas que pueda desplegar la terminal y el máximo de localidades será el numero max. de líneas multiplicado por dieciseis. En una terminal normal se puede llegar a ver una pàgina y media de memoria listada en la pantalla.

Ejemplo:

```
-L109,120 [CR]
-0109 DD DD DD DD DD DD DD
-0110 01 02 02 02 00 00 00 00 00 00 00 00 00 00 00 0C
-0120 12
```

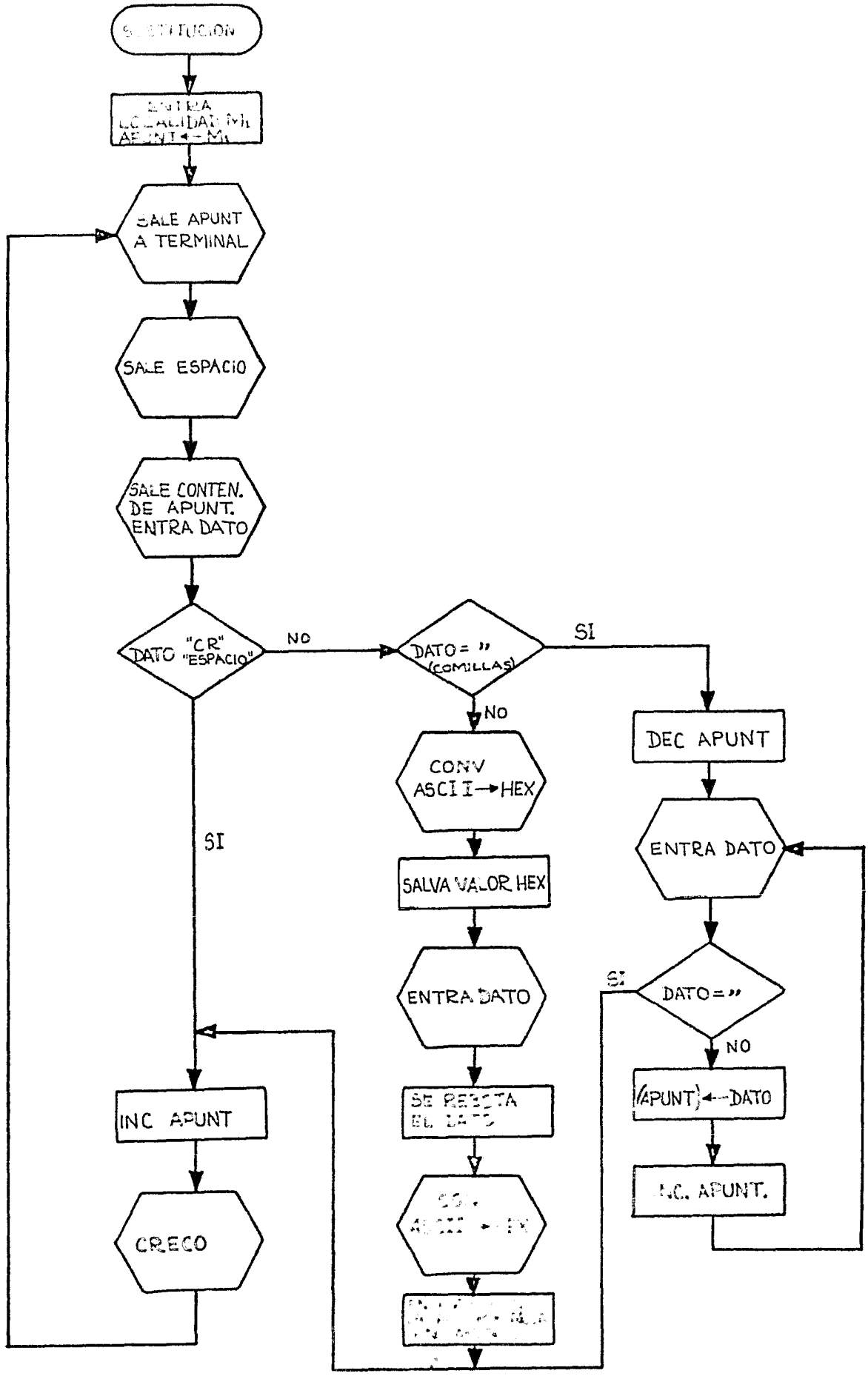
Antes de continuar quiero aclarar algo sobre la forma como se pasan paràmetros a la mayorìa de los comandos; esto siempre y cuando sean necesarios. Para pasar una localidad de memoria como paràmetro debemos saber que ésta entra al programa a través de C016 y es esta rutina la que determina como serà la sintaxis para los comandos.La rutina C016 puede terminarse con un 'CR','LF',',(coma)','espacio' y con el caracter ':'.Por esta razòn podemos utilizar cualquiera de éstos caracteres como delimitador entre parámetros y para que se empiece a ejecutar un comando.De esta manera podemos ver que el comando anterior también lo podemos escribir:

```
-L109:120[espacio]
-L109[espacio]120[CR]
-L109,170:
```

Es recomendable que los parámetros se separen por

"espacio","coma",";" y que después del ultimo parámetro se de un "CR" para que proceda a correr el comando. Si no es aplicada esta recomendación no se afecta el desarrollo del comando siempre y cuando se cumpla con los requisitos de los caracteres delimitadores.

COMANDO DE SUSTITUCION: Este se llama con el caracter 'S' y se le pasa como parámetro la localidad desde la cual empezaremos a sustituir memoria, ya sea caracteres ASCII o valores hexadecimales. Luego de pasarle la localidad inicial aparecerá esta localidad y junto a ella su contenido en hexa. Si deseamos sustituir, solo tenemos que teclear los digitos hexa. que vamos a sustituir, y si no deseamos ningún cambio oprimimos "CR" o "espacio". Después de cualquiera de las operaciones anteriores aparecerá la siguiente localidad con su contenido. Desde aqui la operación se repite hasta que oprimamos una tecla invàlida (caracter no hex. diferente de "CR" o "espacio"). Lo anterior es para sustitución de valores hexadecimales, ahora bien si deseamos sustituir por caracteres ASCII, haremos lo siguiente: luego que aparezca la localidad y su contenido, oprimimos la tecla "'" (comilla) la que indicará que todos los caracteres que se envien desde la terminal se irán sustituyendo en memoria como caracteres ASCII) hasta que se envie otra "'" y se podrá seguir sustituyendo valores hexadecimales.


SALE APUNT A TERMINAL
SALE ESPACIO
SALE CONTEN. DE APUNT. ENTRA DATO
DATO "CR" "ESPACIO"
NO
DATO = " (COMILLAS)
SI
No
SI
CONV ASCII → HEX
SALVA VALOR HEX
ENTRA DATO
DEC APUNT
ENTRA DATO
DATO = "
SI
NO
(APUNT) ← DATO
INC. APUNT.
INC APUNT
CRECO

Ejemplo:

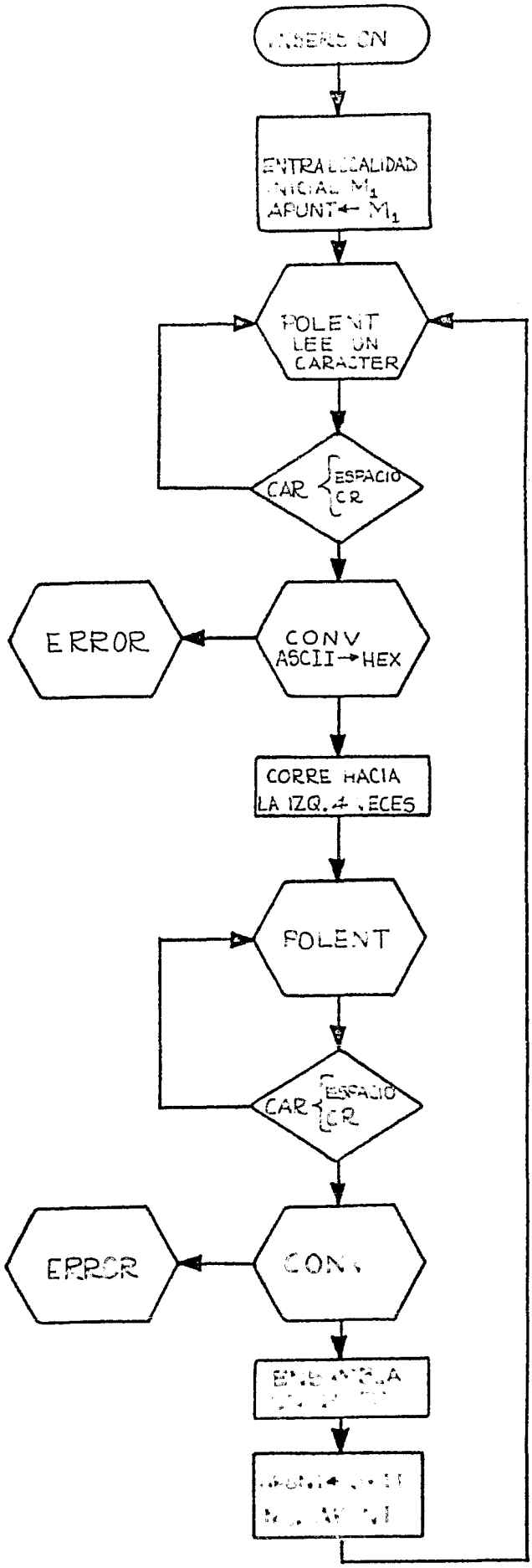
```
        -S1200[CR]          llamada del comando
        -1200 0A 12         sustituye (1200) con 12hex.
        -1201 00 [CR]       no afecta la 1201
        -1202 00 '1234'     sustituye a partir de 1202
                           con '1','2','3','4' ASCII
        -1206 00 78         sustituye (2106) con 78hex.
        -1208 S*            salida del comando
        -
```

INSERCION: éste comando es llamado con el caracter "I", éste comando es muy similar a el de sustitucion y en realidad tiene la misma funciòn pero la realiza de otra manera. Este comando permite solo la inserciòn en memoria de caracteres ASCII correspondientes a valores hexadecimales. Este comando se ejecuta dando la localidad inicial de inserciòn y a continuaciòn una cadena de caracteres correspondientes a valores hexadecimales. Aquì no se despliega ni la localidad donde se està insertando ni el contenido de las localidades de memoria. Es muy ùtil cuando se requiere introducir el còdigo de un programa por primera ocasiòn, ya que ahorra tiempo porque para sustituir el contenido de una localidad y la siguientes no hay que introducir ningun caracter y para terminarlo solo se oprime un caracter invalido en la terminal. En el caso de oprimir "espàcio","CR" o "LF" el comando los desecharà y no afectarà en nada su operaciòn.

Ejemplo:

```
        -I[illegible][CR]
        -[illegible]DA [illegible]134 [illegible]41[illegible] CDDFFF [illegible]540*
        -
```

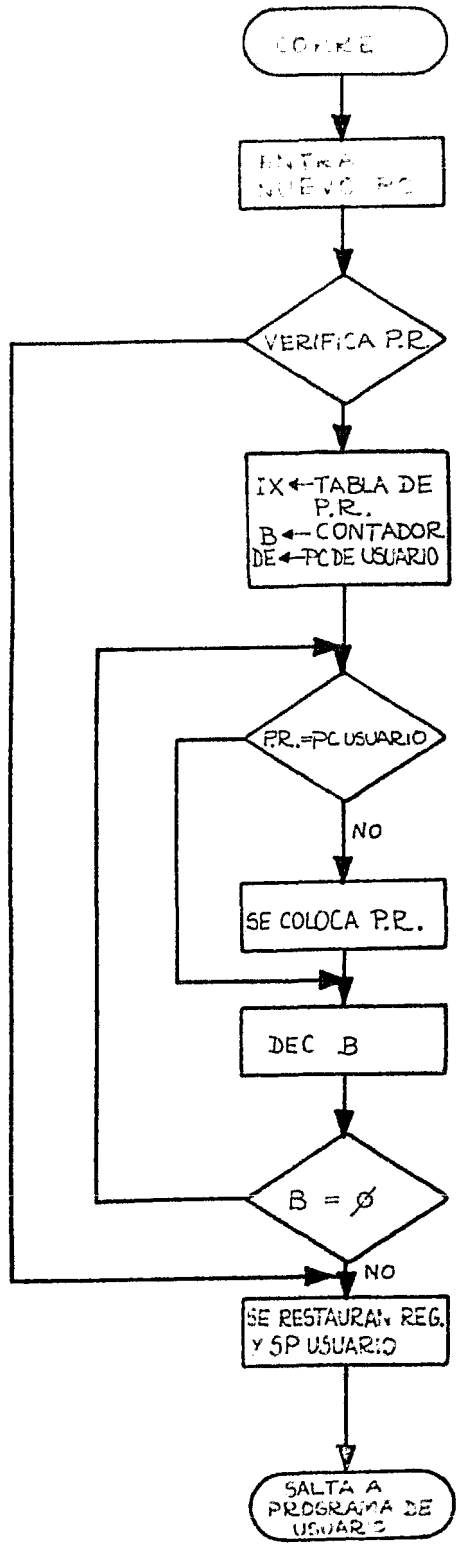
con el caracter "*" nos salimos en éste caso del comando

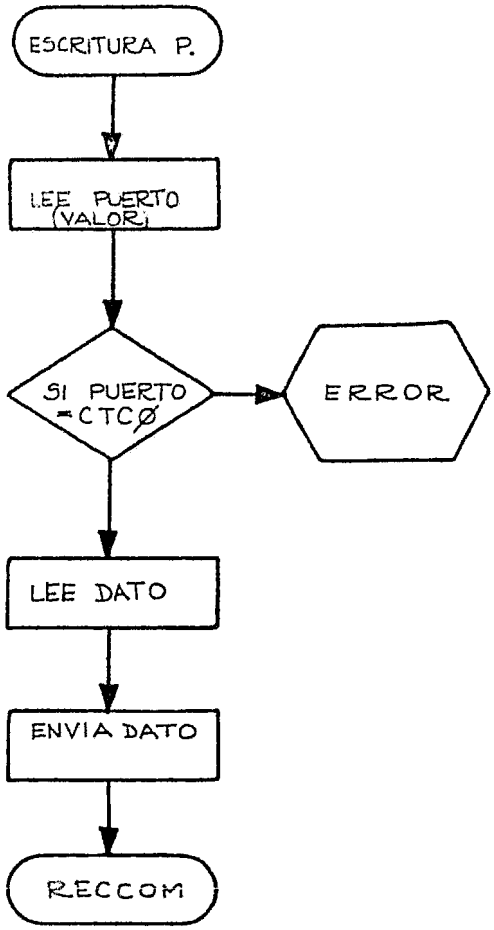

ENTRA LOCALIDAD
APUNT ← M1
POLENT
LEE UN
CARACTER
CAR { ESPACIO
CR
ERROR
CONV
ASCII → HEX
CORRE HACIA
LA IZQ. 4 VECES
POLENT
CAR { ESPACIO
CR
ERROR

COMANDO CORRE: éste comando es llamado con el caracter "C" y nos permite correr programas que hayamos insertado en la memoria Ram del equipo. A éste comando le damos como paràmetro la localidad donde empieza el còdigo de nuestro programa y él restaurarà los valores de los registros que han sido salvados por la rutina RST8, verificarà los puntos de ruptura para el programa y los colocarà; por último enviarà el control del programa a nuestro programa.

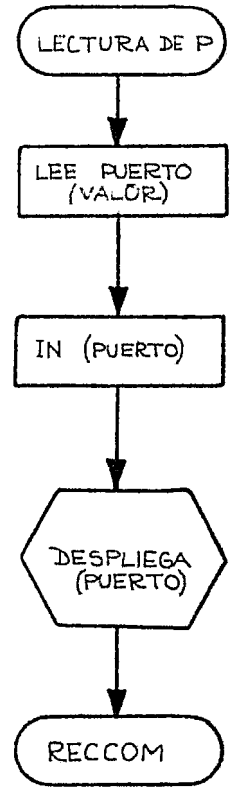
COMANDO LECTURA DE PUERTO: éste comando es llamado con el caracter "E" y nos permite leer el contenido de un puerto y desplegarlo en valor hexadecimal en la terminal. Hay que recordar que si deseamos leer datos del Usart con éste comando habrà que hacer una pequeña modificaciòn ya que éste comando solo lee el puerto que le pasemos como paràmetro y en éste caso particular cuando se lea el buffer de datos, encontraremos el último dato que llego al Usart y éste serà caracter con que hacemos que se ejecute el comando.

COMANDO ESCRITURA DE PUERTO: éste comando es llamado con el caracter "O" y nos permite enviar un valor hexadecmal a cualquier puerto conectado a nuestro sistema. Como proteccion particular, no se permite mediante éste


VERIFICA P.R.
IX←TABLA DE P.R.
B←CONTADOR
DE←PC DE USUARIO
P.R.=PC USUARIO
NO
SE COLOCA P.R.
DEC B
B = ø
NO
SE RESTAURAN REG. Y SP USUARIO
SALTA A PROGRAMA DE USUARIO



COMANDO ESCRITURA DE PUERTO



COMANDO LECTURA DE PUERTO

comandos el envio de datos al canal cero del CTC ya que nos quedariamos sin comunicaciòn si alteramos la operaciòn de éste canal.

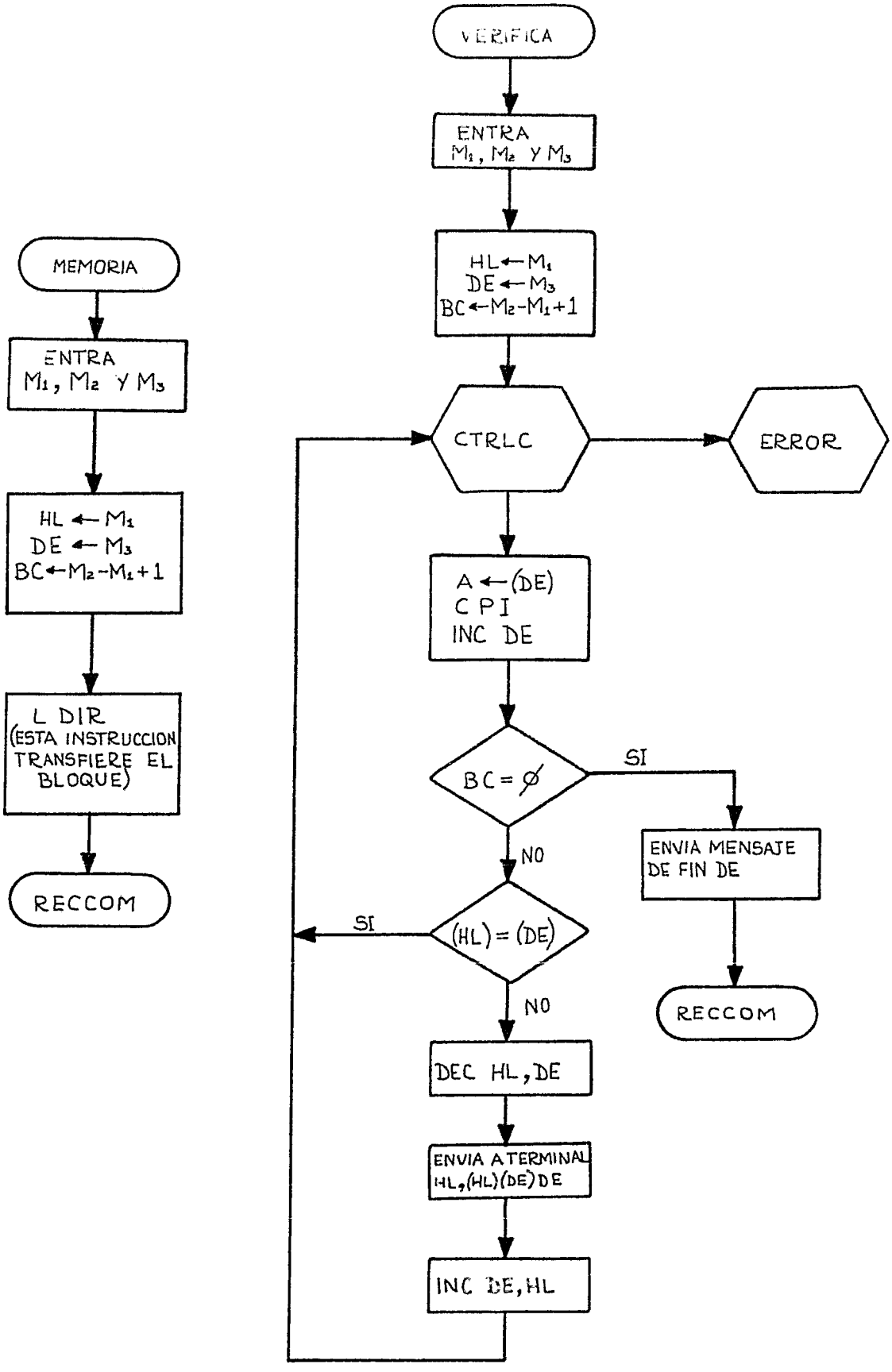
Ejemplos de los dos comandos anteriores:

```
-E45 "0F"
-E87 "56"
-098  50[CR]
-064 *
-
```

el caracter "E" es para lectura y las comillas nos indican que no es escrito por nosotros. En el ùltimo ejemplo se quiso enviar un dato al canal cero del CTC.

COMANDO MEMORIA: éste comando es llamado con el caracter "M" y nos permite mover bloques de memoria, dando como paràmetros dos localidades que definen el bloque a mover y una tercera localidad a donde se moverà. Todo esto es muy fàcil gracias a la instrucciòn LDIR del mP Z80 la cual permite mover bloques de memoria con solo cargar los reg. HL, BC y DE con ciertos paràmetros.

COMANDO DE VERIFICACION:éste comando es llamado con el caracter "V" y nos permite comparar dos zonas de memoria de nuestro sistema. Este comando es de gran ayuda cuando se programan memorias EPROM ya que podemos ver si se grabò correctamente. En el caso de que las dos zonas sean idénticas en su contenido solo se desplegarà en la


MEMORIA
ENTRA M1, M2 Y M3
HL ← M1
DE ← M3
BC ← M2-M1+1
L DIR
(ESTA INSTRUCCION TRANSFIERE EL BLOQUE)
RECCOM
VERIFICA
ENTRA M1, M2 Y M3
HL ← M1
DE ← M3
BC ← M2-M1+1
CTRLC
ERROR
A ← (DE)
C P I
INC DE
BC = Ø
SI
NO
ENVIA MENSAJE DE FIN DE
RECCOM
(HL) = (DE)
SI
NO
DEC HL, DE
ENVIA A TERMINAL HL, (HL) (DE) DE
INC DE, HL

terminal el mensaje -FC, (fin de comparación). En caso de que se encuentren localidades con diferente contenido, se desplegaran las dos localidades con sus respectivos contenidos.

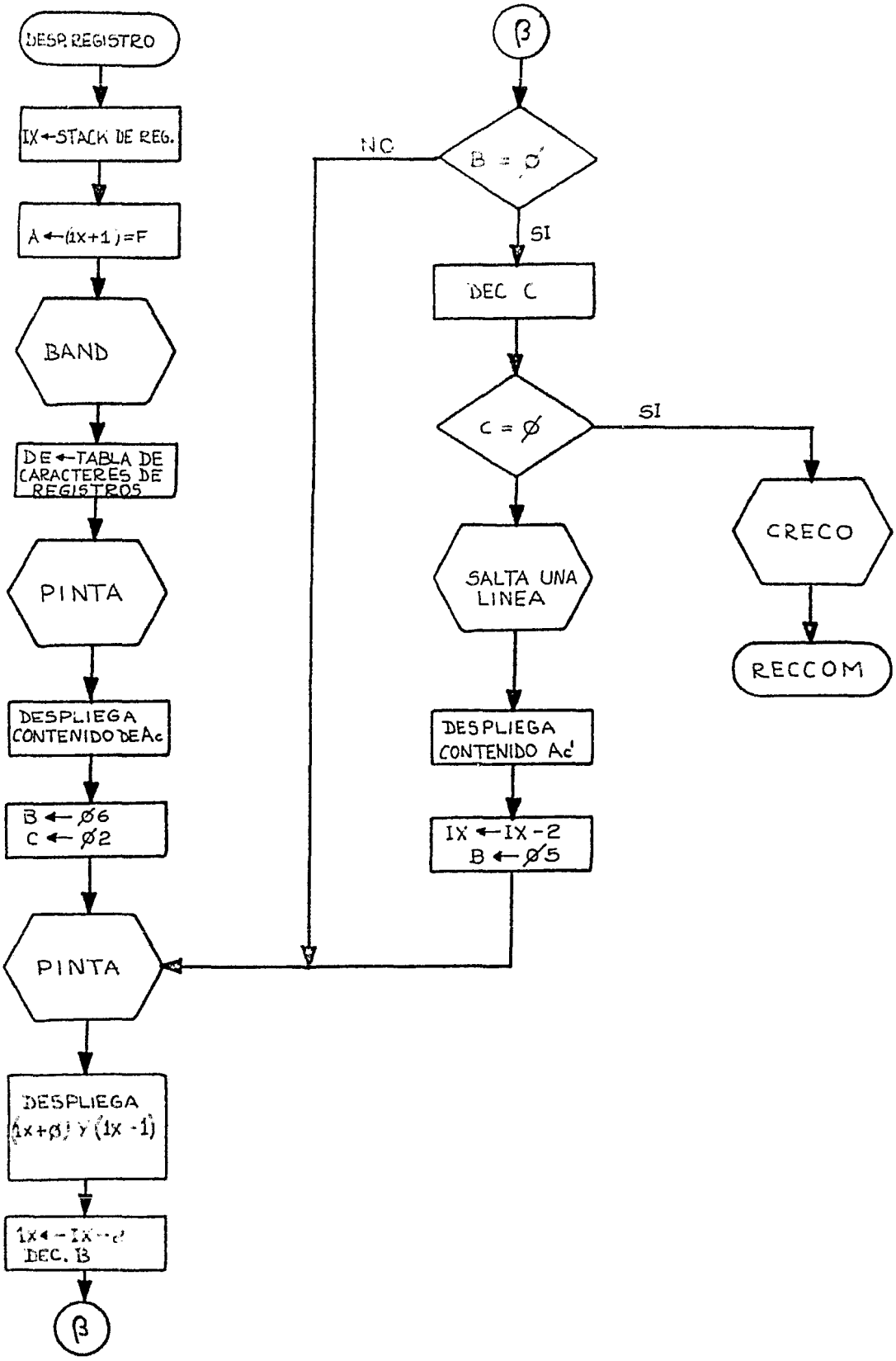
Ejemplo:

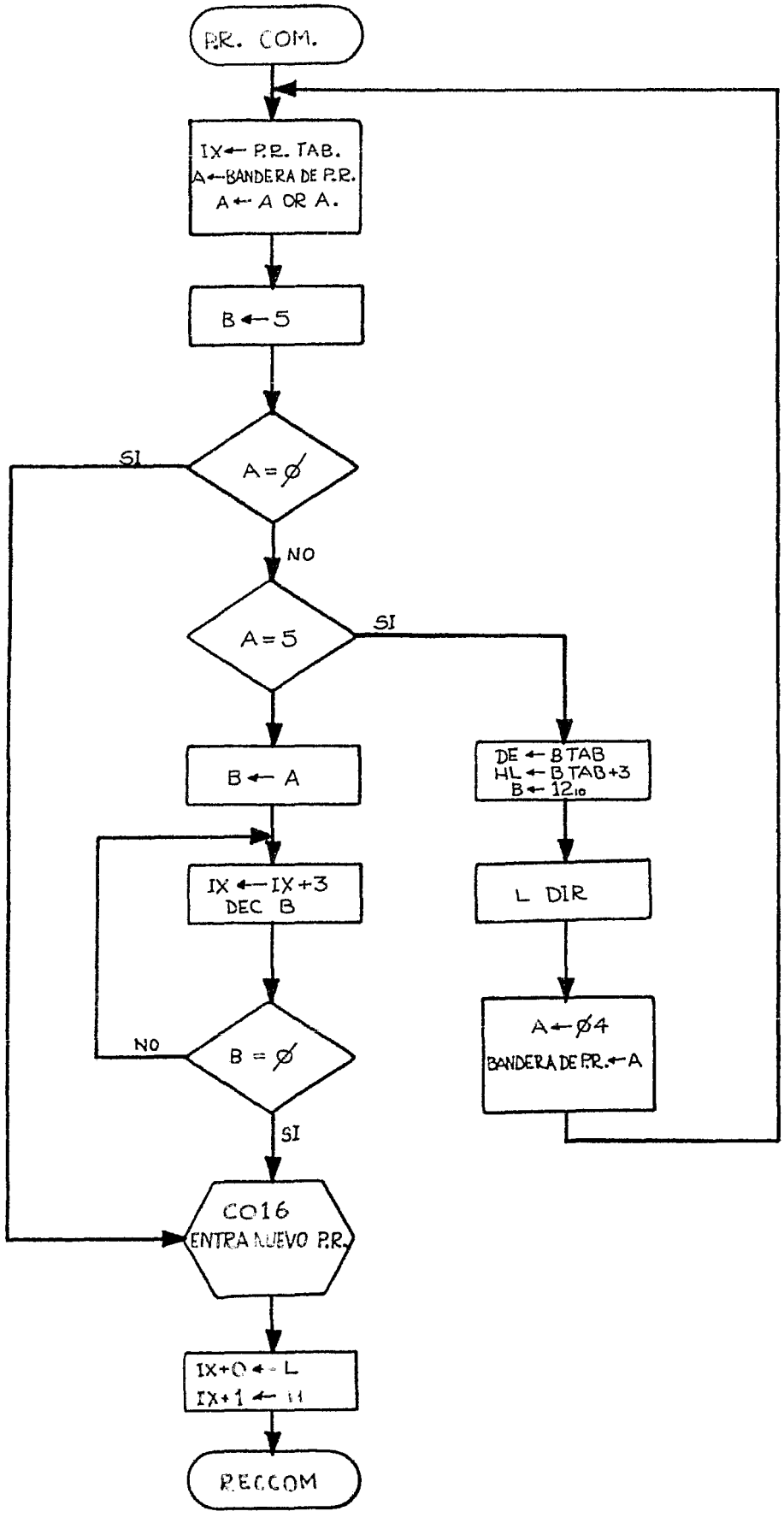
```
-V00,200,1000
-FC.
-V100,200,2000
-100 23 45 2000
-101 65 67 2001
-1F5 98 00 20F5
-FC.
```

en el primer caso se verifico de la 0000hex a la 200hex con la 1000hex hasta la 1200, sin encontrar ningùna diferencia; mas en el segùndo ejemplo se encontraron diferencias en las localidades 100,101,1F5 con la 2000,2001,20F5 respectivamente.

COMANDO DE REGISTROS: éste comando es llamado con el caracter "R" y permite desplegar el contenido de los registros del mP. A éste comando no necesitamos pasarle paràmetros. Llamara a las rutinas BAND (para desplegar las banderas del mP) y PINTA para pintar en la pantalla los diferentes registros.

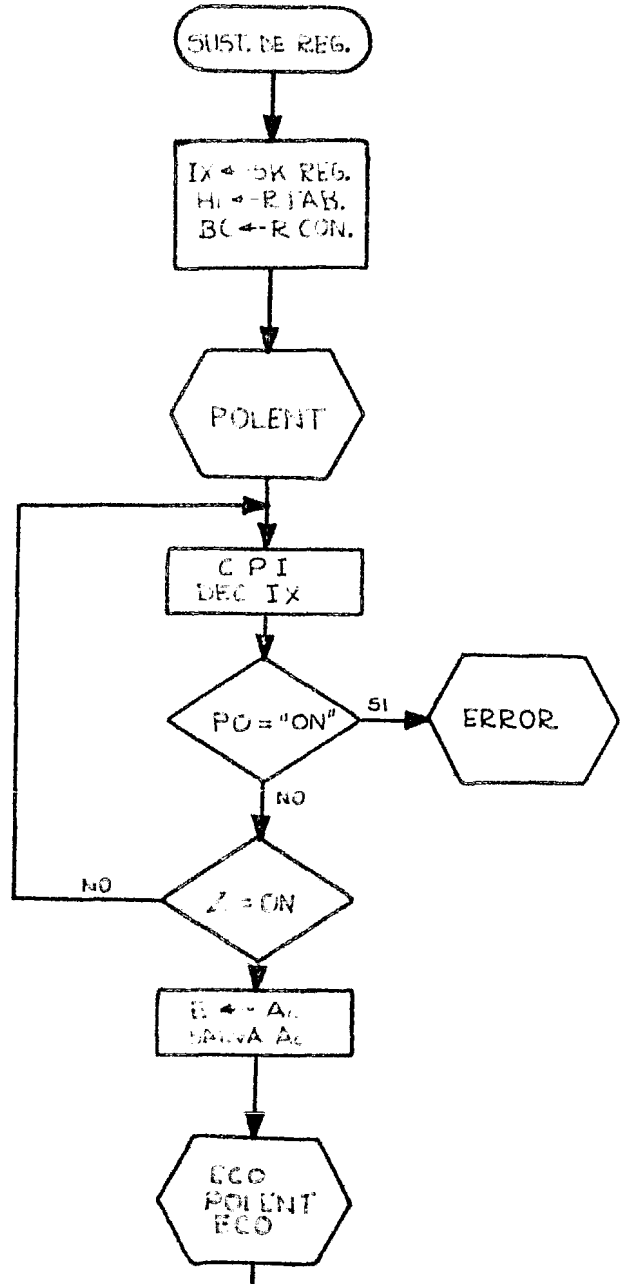
COMANDO RUPTURA: Este comando es llamado con el caracter "B" y nos permite colocar puntos de ruptura en nuestros programas. Para esto pasamos como parametro la localidad donde queremos [illegible] para cancelar [illegible] este comando

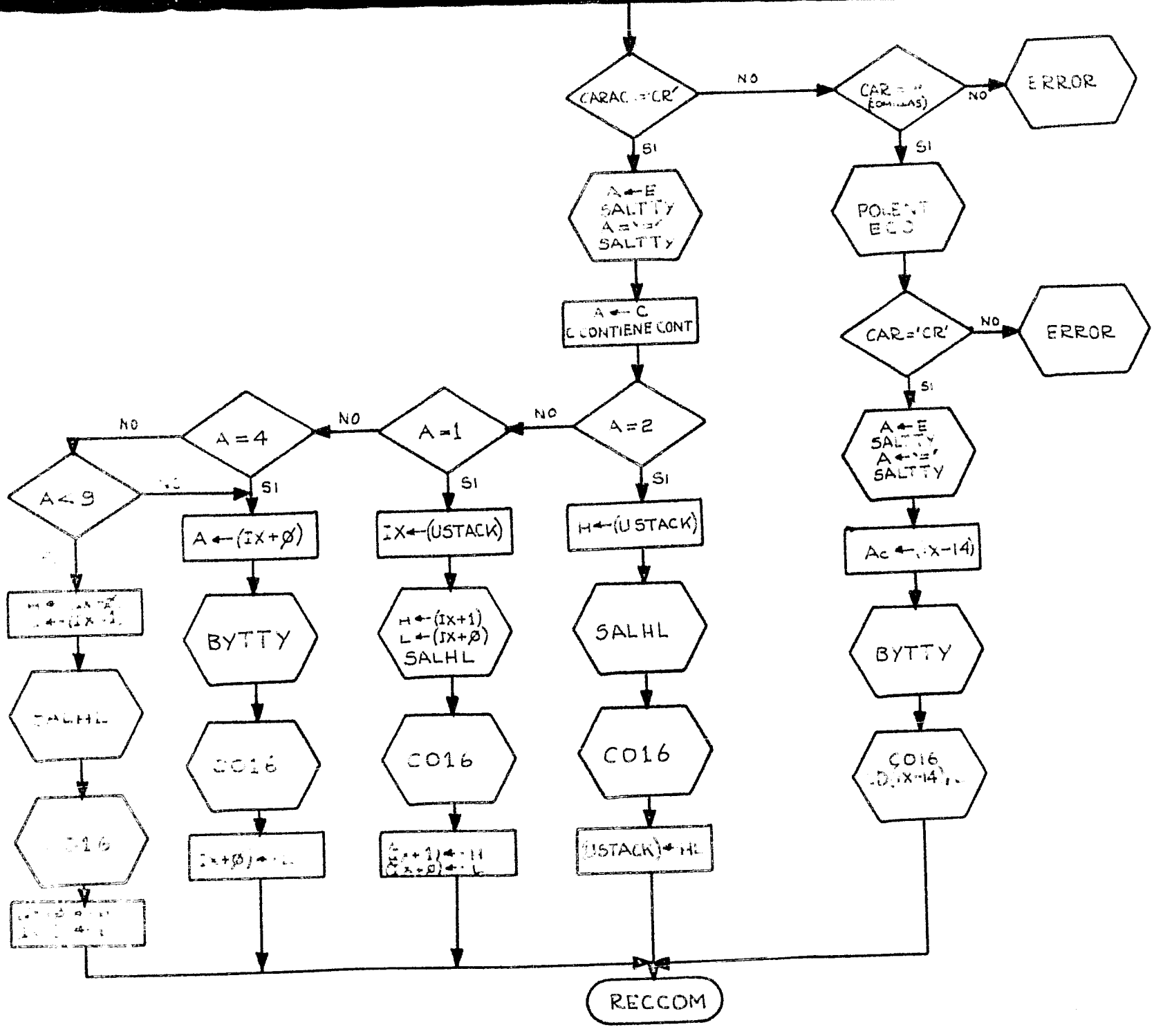

DESP REGISTRO
IX←STACK DE REG.
A←(IX+1)=F
BAND
DE←TABLA DE CARACTERES DE REGISTROS
PINTA
DESPLIEGA CONTENIDO DE Ac
B←Ø6
C←Ø2
PINTA
DESPLIEGA (IX+Ø) Y (IX-1)
IX←IX-2
DEC. B
β
β
B = Ø
NO
SI
DEC C
C = Ø
SI
SALTA UNA LINEA
DESPLIEGA CONTENIDO Ac'
IX←IX-2
B←Ø5
CRECO
RECCOM


P.R. COM.
IX← P.R. TAB.
A←BANDERA DE P.R.
A← A OR A.
B←5
A=Ø
SI
NO
A=5
SI
B← A
IX←IX+3
DEC B
B=Ø
NO
SI
DE ← B TAB
HL ← B TAB+3
B← 12₁₀
L DIR
A←Ø4
BANDERA DE P.R.←A
CO16
ENTRA NUEVO P.R.
IX+0 ← L
IX+1 ← H
REGCOM

solo guardara esta localidad en una tabla especial para esto y el contenido será procesado por el comando Corre y la rutina RST8. En la tabla tendremos tres localidades por punto de ruptura que coloquemos; las dos primeras para la localidad donde se colocará y la tercera para salvar el contenido de esa localidad mientras corre el programa y sustituir esa localidad con OCFhex., que corresponde a la instrucción RST8. Este comando es similar al del Starter Kit con la diferencia que la entrada es por terminal y que no se colocaran los puntos de ruptura cuando la localidad sea la misma que la que introduzcamos con el comando Corre.

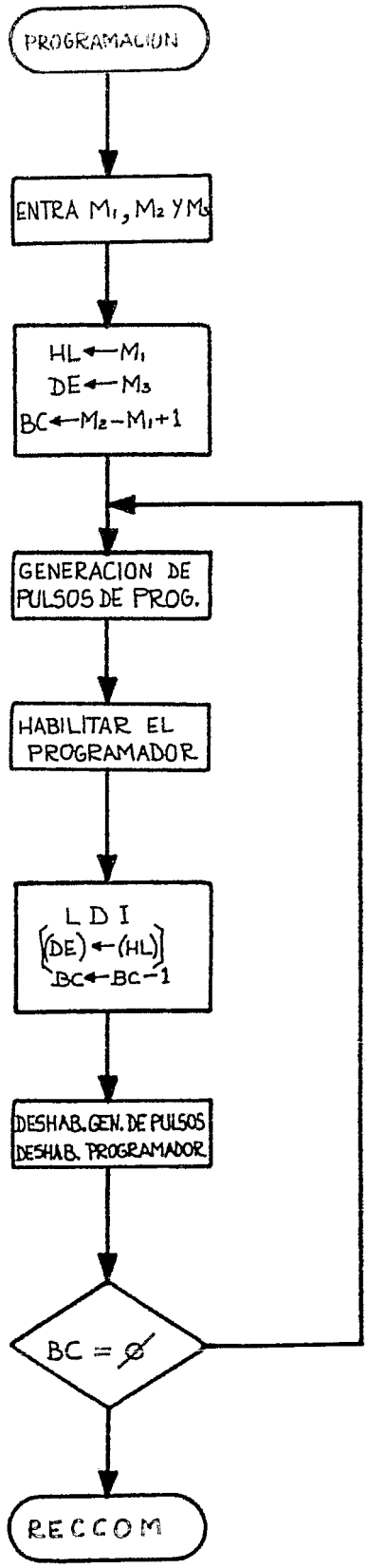
COMANDO DE SUBST. DE REGISTROS: éste comando es llamado con el caracter "R" y permite cambiar el contenido de los registros internos del mP. Como en realidad cuando desplegamos el contenido de registros solo vemos los contenidos que fueron salvados en memoria por la rutina RST8 y no el contenido actual, éste comando se limita a sustituir localidades de memoria en realidad. Claro que las localidades de los registros están bien determinadas en nuestro sistema y es por esto que se puede hacer la sustitución de éstos sin cometer errores. Para realizar la sustitución se cuenta con una tabla con los caracteres correspondientes a todos los reg. internos del mP Z80. Cuando se llama a éste comando se pasa como parámetro el


SUST. DE REG.
IX ← SK REG.
HL ← R TAB.
BC ← R CON.
POLENT
C P I
DEC IX
PO = "ON"
SI
ERROR
NO
Z = ON
NO
B ← A
SALVA A
ECO
POLENT
ECO


NO
NO
ERROR
SI
SI
A←E
SALTTY
A←'='
SALTTY
POLENT
ECO
A←C
C CONTIENE CONT
CAR='CR'
NO
ERROR
SI
A=2
A=1
A=4
A<9
NO
NO
NO
SI
SI
SI
SI
A←E
SALTTY
A←'='
SALTTY
H←(USTACK)
IX←(USTACK)
A←(IX+Ø)
Ac←(IX-14)
SALHL
H←(IX+1)
L←(IX+Ø)
SALHL
BYTTY
BYTTY
CO16
CO16
CO16
CO16
RECCOM

caracter correspondiente al reg. que deseamos sustituir y en caso de ser un reg. prima le seguira el caracter "'". Para los reg. de 16 bits corresponden los siguientes caracteres IX="X",IY="Y",SP="S",PC="P". Una vez que ha sido pasado el caracter correspondiente al reg. aparecerá éste con su contenido actual y con solo teclear el nuevo contenido y un "CR" habremos hecho la sustitución. Si no deseemos la sustitución solo tecleareMos "CR".

COMANDO DE PROGRAMACION:éste comando es llamado con el caracter "P". Este comando particular del Starter Kit fue diseñado por las facilidades que ofrecia el Starter KIt. A éste comando se le pasan los paràmetros igual que al comando de Memoria, con la única diferencia que la tercera loacalidad que se de debera estar dentro de la memoria EPROM. Para éste comando solo necesitamos habilitar un puerto para habilitación de unos pulsos que van a la entrada "Wait" del mP y que permiten que los datos a grabar en la EPROM estén presentes el tiempo suficiente en el bus de datos para que puedan ser grabados. Para la operación de grabación de memoria deberemos contar con una fuente adicional de +25volts y además poner el interruptor S2 en Pgrm.


PROGRAMACION
ENTRA M1, M2 Y M3
HL ← M1
DE ← M3
BC ← M2 − M1 + 1
GENERACION DE PULSOS DE PROG.
HABILITAR EL PROGRAMADOR
LDI
(DE) ← (HL)
BC ← BC − 1
DESHAB. GEN. DE PULSOS
DESHAB. PROGRAMADOR
BC = Ø
RECCOM

Sistemas de Desarrollo Para Microprocesadores.

Usualmente se asocia el término de sistema de desarrollo a los Sistemas de Desarrollo de Interconexiòn por Bus. En éstos se denomina Bus al conjunto de lineas que llevan señales de control, informaciòn y energia al sistema. En éste caso el proyecto con Microprocesador se conecta al bus del sistema, recibiendo asi las señales del sistema de desarrollo; para realizar esta interconexiòn es necesario que se conozcan y respeten los protocolos que utiliza el procesador del sistema de desarrollo ya que éste actùa también como controlador del bus y si deseamos que nuestro proyecto tenga acceso al bus debemos hacer la interconexiòn con sumo cuidado. En éstos sistema los dispositivos que estàn conectados al bus sensan las señales que se presentan en éste para reconocer si se està solicitando dicho dispositivo o si debe dejar el bus libre. Ahora bien para que dos microprocesadores estén conectados al mismo bus uno de los dos será el que controle a dicho bus en tiempos determinados. De esta forma el procesador del sistema de desarrollo puede [illegible] que se encuentren dentro del proyecto en [illegible] el procesador del sistema en desarrollo [illegible] de alta impedancia.

Esto es de gran ventaja porque se pueden hacer pruebas de dispositivos con un solo procesador (el del sist. de desarrollo).

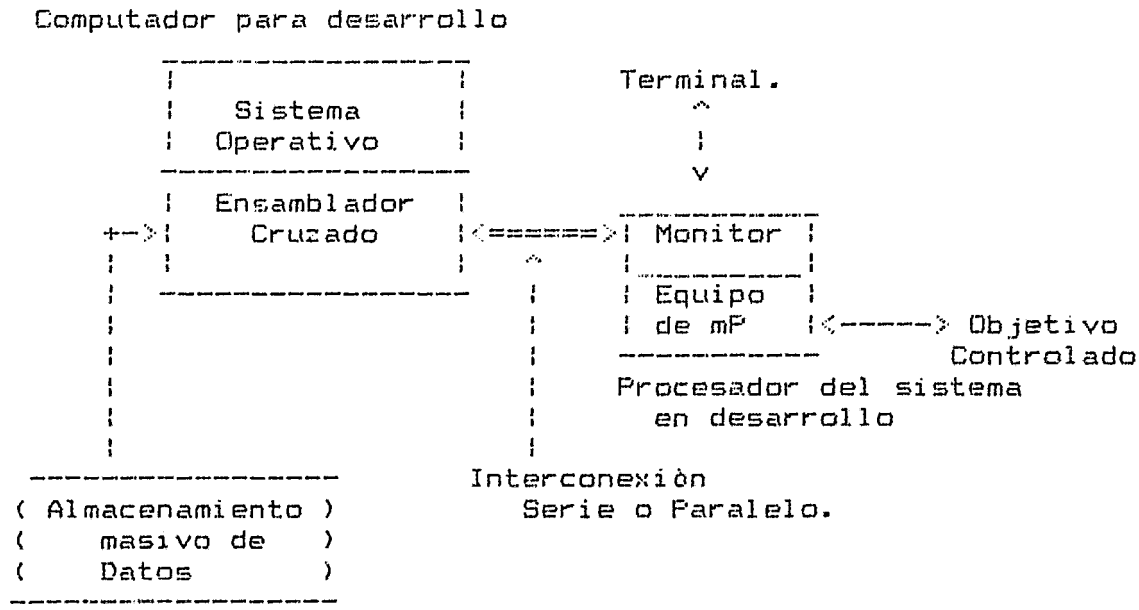
Sistemas de Desarrollo por Descarga o Vaciado.

En éstos sitemas de desarrollo por lo general el procesador del sistema de desarrollo es totalmente independiente del procesador del proyecto en desarrollo . En éstos sitemas contamos con un pequeño equipo, con el procesador en estudio para realizar nuestro proyecto; éste equipo deberà tener algunos puertos destinados al proyecto. Por otro lado contamos con un computador que posee las ventajas de lenguajes de alto nivel, editores de texto y programas para desarrollo de proyectos de micros. Uno de éstos programas es por lo general un "Ensamblador Cruzado", éste tipo de ensamblador genera còdigo para una màquina diferente a la que corre. Para que el sistema quede completo se necesita un lazo entre el equipo y el computador; esta uniòn se realiza mediante un programa de transferencia que reside en el computador y mediante el cual vaciamos programas a nuestro equipo.

Dadas las caracterìsticas de éste tipo de sistemas decidimos crear un sistema de desarrollo por descarga formado por el Starter-kit Z80 y el computader PC de Promenco. Para esto solo se necesitaba un tipo de conexiòn entre los dos equipos

y el programa de transferencia. El enlace fué relativamente sencillo ya que se había terminado el monitor para terminal y solo hubo que desarrollar el programa de transferencia.

Diagrama de un Sistema de Desarrollo por Descarga



Sistema Cromenco S-2

Una de las diferencias basicas de los programas en ensamblador con los programas de alto nivel estriba en que al programar en nivel ensamblador se tiene que tener conocimiento del "ambiente" que rodea al mP. Cuando se programa en nivel ensamblador hay que tomar en cuenta las entradas y salidas del sistema, tal y como son en la mayoria de los casos(puertos) ademàs que se trabaja a un nivel elemental donde la mayoria de las utilidades del sistema se tendràn que implementar. En muchos casos, al trabajar en nivel ensamblador debemos tomar en cuenta la configuraciòn de hardware y software del sistema(capacidad de memoria, interrupciones al sistema y puertos de entrada y salida). Asi como nos es de mucha ayuda en un proyecto de harware, un analizador de estado, asi también, es de gran ayuda un Ensamblador para proyectos de software. Ya que este proyecto para mP Z80 es de software en general, la parte correspondiente al hardware serà explicada posteriormente por su sencillez.

Sistema operativo CP/M

Primero que todo deseo explicar brevemente algo sobre el

sistema operativo del sistema Cromenco S2. Se lo denomina CDOS (Cromenco Disk Operation System) y es una versiòn aumentada y mejorada del popular sistema operativo de disco CP/M que corre en sistemas basados en mP 8080 ,mP 8085 y por supuesto en el mP Z80. Este sistema operativo fue creado originalmente por ingenieros de la compañia Intel para sistemas basados en el bus S-100 del 8080 y discos flexibles de 8 pulgadas. Actualmente las versiones de CP/M pueden correr en otras configuraciones que no posean el bus S-100 como en la TRS 80 y en sistemas con disco de 5 pulgadas como los del sistema Cromenco S2. Ademas existe una version multiusuario denominada MP/M.

CP/M es un tipo de software que puede ser corrido en diferentes configuraciones de computadora (con ciertas limitaciones) a travez de un conjunto de rutinas de interface. Una vez que el sistema es cargado a la màquina, los programas que corran con CP/M seràn independientes de la maquina o sistema. Hace algunos años (en mComputadoras) cuando no se utilizaba CP/M cada ensamblador o intérprete de Basic debìan contar con su propio editor de texto; con CP/M la ediciòn es un proceso independiente de la ejecuciòn del programa, es CP/M se utiliza un editor independiente del sistema para crear un archivo en codigo ASCII el cual es utilizado como archivo fuente por un compilador o un ensamblador. Para nuestro caso, despuès de ensamblado deberà ser [illegible] archivo para que pueda ser ejecutado en el

sistema. El sistema CP/M tiene una sola dirección de memoria(0005) para entrada a sus operaciones. Cuando se hace una llamada a esta dirección, el registro C del mp deberá contener el número correspondiente a la operación que se realice y los parámetros que se necesiten serán pasados a traves de los otros registros del mP. Hay un total de 37 rutinas diferentes del sistema que pueden ser llamadas de la forma que se mencionó. Las operaciones de la rutina 0 a la 11 están orientadas a entrada y salida de datos del sistema y relacionadas a los cuatro periféricos cuyos nombres lógicos coresponden a: consola, lista(impresora), perforadora y lectora. En el sistema operativo del sistema Cromenco S2 CDOS se cuenta con un total de 160 rutinas diferentes. Para el proyecto solo se utilizaron algunas de las rutinas de entrada y salida, para que el programa pudiese correr en el sistema. Esto asegura que el programa puede correr en cualquier sistema que tenga un sistema operativo CP/M y un mP Z80 como CPU.

Para el programa especial de transferencia de archivos ensamblados de la Cromenco al equipo de entrenamiento se utilizaron rutinas especiales de una biblioteca del ensamblador. La mayor parte de estas rutinas están relacionadas con el manejo de archivos y conversion de bases numericas. Para poder utilizar las rutinas de esta biblioteca es necesario conocer algo sobre el control de archivos en CP/M. Para poder recomocer un archivo en un disco, el sistema

operativo debe manejar un nombre simbólico, dado por el usuario. Este nombre simbólico se coloca en un area de memoria denominado "Bloque de Control de Archivo" el cual esta constituido por 33 bytes, en este BCA se especifca el drive donde se encuentra, nombre, tipo o extenciòn y otras caracterìsticas del archivo.

Macroensamblador Z80

El Macroensamblador Z80 genera el còdigo objeto correspondiente a los mnemònicos de un programa fuente creado en un editor de texto. Este macroensamblador presenta las siguientes facilidades: se pueden generar macros para funciones comunes en el programa, se pueden escribir bloques que sean ensamblados en relaciòn a una condiciòn, se pueden crear programas que sean relocalizables o de còdigo absoluto y se pueden escribir programas muy grandes utilizando la facilidad de "mòdulos" que son ensamblados independientemente y luego ligados. Este Macroensamblador corre bajo el sistema operativo de Cromenco CDOS.

Para realizar un programa en ensamblador, se crea una archivo en un editor de textos el cual servirà como fuente para el Macroensamblador (mnemonicos) y se denomina de tipo ASM. El Macroensamblador, puede dar como salida tres tipos diferentes de archivos, el primero que es por default es de

tipo relocalizable que debera ser ligado para poder ser ejecutado; se denomina de tipo REL. El segundo es opcional y es un listado del proceso de ensamblado; se conoce com tipo PRN, el tercero también es opcional y es un archivo especial en caracteres ASCII de números hexadecimales y es un formato conocido como Intel (para archivos). Puede ser corrido en el programa simulador Debugger y puede servir si se desea mandar grabar una memoria ROM. Este último tipo de archivo se denomina tipo HEX.

Nuevas Instrucciones Para el Z80

Gracias a la ventaja de utilizar un Macroensamblador se pueden crear mnemònicos para instrucciones que no han sido documentadas por Zilog, quien produce el mP Z80.

Para las personas que han utilizado el Z80 los registros IX e IY siempre son utilizados como registros de 16 bits, pero también pueden ser utilizados como dos registros de 8 bits cada uno. Para esto debemos saber que Zilog tuvo que agregar un byte màs por instrucciòn para los registros índice; debido a esto para el registro IX tendremos un byte DDhex como el primer byte del còdigo de operaciòn, seguido de éste viene el byte que indicarà la operaciòn a realizar, pero este segundo byte es igual al còdigo para una instrucciòn del par de regitros HL. Un ejemplo de esto es :

Ya ensamblado

21 00 12 LD HL,1200H
DD 21 00 12 LD IX,1200H

Ahora bien si a códigos para instrucciones de los registros H y L por separado, le antecedemos el codigo DDhex estaremos realizando la misma instruccion como si fuera el registro L con los 8 bits menos significativos del reg IX y en caso de ser una instrucciòn para H realizaremos la misma instrucciòn para los 8 bits màs significativos de IX. De esta manera podemos utilizar nuevas instrrucciones que no fueron documentadas por Zilog.Por ejemplo:

```
    67    LD      H,A      ;cargar el reg. H con A.
DD  67    No hay mnemònico ;carga los 8 bits
                           màs sig. de IX con A.
```

Lo anterior se cumple en forma idéntica para el reg. IY solo que su primer byte es FDhex. Ademàs de las operaciones mostradas en los ejemplos, ésto se cumple para las siguientes instrucciones de H y L:

```
ADC     A,H          ADC     A,L
AND     H            AND     L
CP      H            CP      L
DEC     H            DEC     L
INC     H            INC     L
OR      H            OR      L
SBC     A,H          SBC     A,L
SUB     H            SUB     L
XOR     H            XOR     L
```

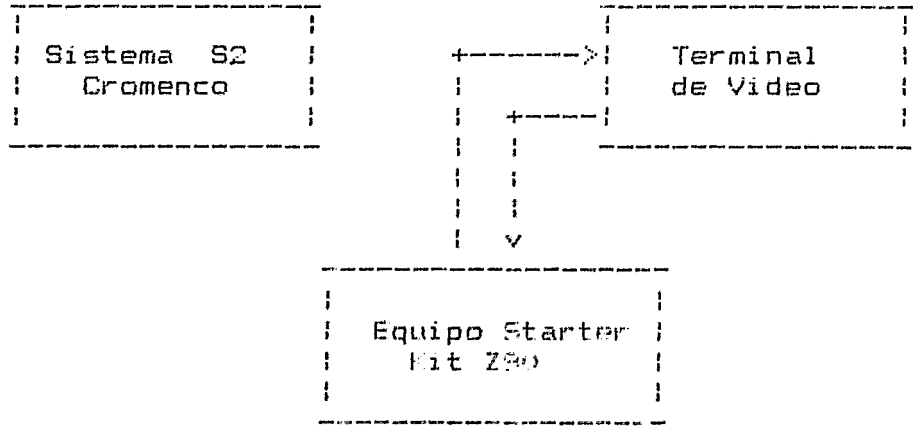
Con el macroensamblador podemos crear nuevos mnemònicos como por ejemplo:

```
        LDXH  (cargar los 8 bits màs significativos de IX)
LDXH    MACRO   Reg
        DB      0DDH
        LD      H,Reg
        ENDM
```

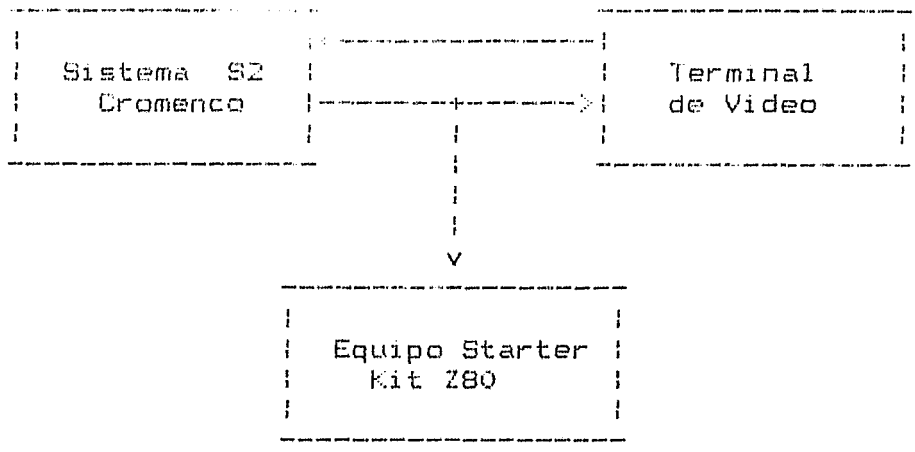
TRANSFERENCIA DE INFORMACION E INTERFACE
DEL SISTEMA DE DESARROLLO POR VACIADO

Como se menciono con anterioridad, nuestro sistema de Desarrollo fué de facil implementaciòn una vez que el programa monitor habia sido concluido. Ahora veremos las dos posibles configuraciones que podemos tener del sistma de desarrollo respecto a los componentes y el flujo de informaciòn:

Configuraciòn A: en ésta el equipo Cromenco no tiene comunicaciòn con ninguno de los otros elementos que forman el sistema. Esta configuraciòn implica comunicaciòn bidireccional entre el Starter kit y la terminal.



Configuraciòn B: en ésta el equipo Cromenco tiene comunicaciòn bidireccional con la terminal y ademàs el equipo Starter Kit recibe la misma informaciòn que le llega a la terminal.



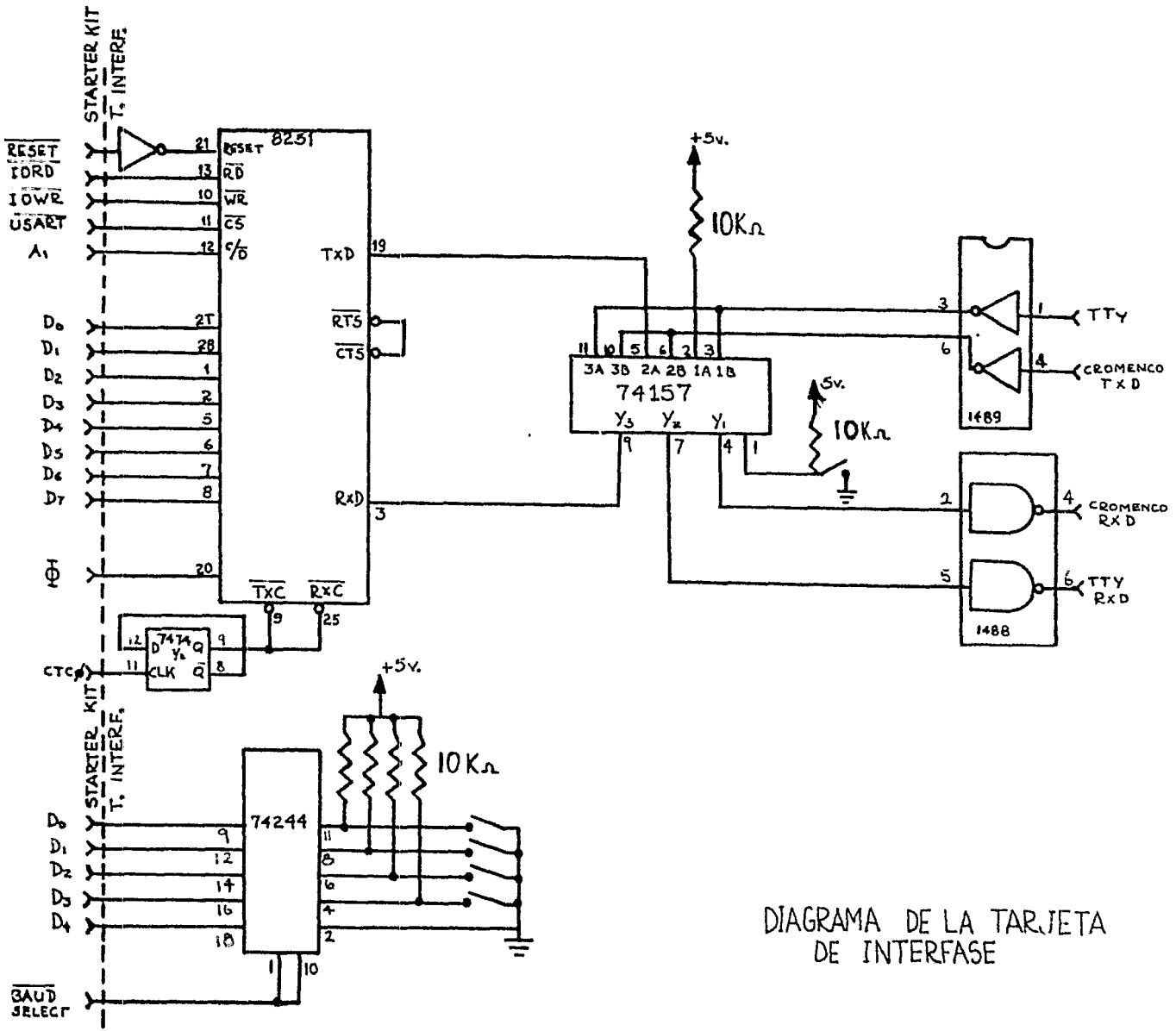
Interface de los Equipos.

La conexiòn de los tres equipos se realizo mediante el circuito mostrado en la figura. Para poder cambiar de una configuraciòn a otra se utilizo un C.I. 74157 el cual sirvio para implementar tres funciones booleanas las cuales representan las entradas de informaciòn a cada uno de los equipos:

Starter(ent)= (Cromenco(sal) And Cr) + (Terminal(sal) And Cr')
Terminal(ent)= (Cromenco(sal) And Cr) + (Starter(sal) And Cr')
Cromenco(ent)= (Terminal(sal) And Cr) + Cr'

Donde Cr. es una variable de control.
Para Cr="0" estaremos en la configuraciòn A
y para Cr="1" en la B.

Ademàs se utilizaron los circuitos 1488 y 1489 los cuales sirven para trabajar segùn la norma para comunicaciòn: RS-232.


STARTER KIT T. INTERF.
RESET
IORD
IOWR
USART
A1
D0
D1
D2
D3
D4
D5
D6
D7
Φ
CTC
8251
RESET
RD
WR
CS
C/D
TxD
RTS
CTS
RxD
TxC
RxC
+5v.
10KΩ
74157
3A 3B 2A 2B 1A 1B
Y3 Y2 Y1
5v.
1489
TTY
CROMENCO TXD
1488
CROMENCO RXD
TTY RXD
74244
BAUD SELECT
DIAGRAMA DE LA TARJETA DE INTERFASE

Quizas la parte esencial en el Sistema de Desarrollo por descarga sea el Programa de transferencia para archivos ensamblados hacia el Starter Kit Z80.

Para realizar la transferencia hacemos uso del comando de Insercion por lo que nuestro comando solo necesitarà proveer la localidad inicial de insercìon al comando y luego enviar el archivo hacia el Starter Kit.

Ya se menciono anteriormente que el Macroensamblador puede dar como salida diferentes archivos. Uno de éstos es el archivo tipo Hex. y no es mas que un archivo de caracteres ASCII segun el formato de Intel.

Formato Intel.

Este formato se escogio ya que el archivo que genera por "default" el Macroensamblador contiene codigo relocalizable y éste, presenta dos problemas: el primero es que habrìa que convertir el còdigo a caracteres ASCII para poder hacer la transferencia y segundo que posee codigos directrices para el ligador, los cuales tendriamos que desechar. A diferencia del archivo relocalizable el formato de Intel presenta la ventaja que la informaciòn del código del programa a transferir esta [illegible] como caracteres ASCII. Por lo [illegible] el programa de

transferencia. Este formato para archivo divide al archivo en registros (de tamaño variable) y se organiza de la siguiente manera:

Byte dentro de un reg.	Utilidad
1	Delimitador de registros caracter ":"
2-3	Numero de bytes en el registro.
4-7	Localidad a partir de donde se Ensamblo el reg.
8-9	Caracteres 00 ASCII.
10-	Datos del registro

Al final del registro hay dos Bytes que son el "cheksum" de los bytes del registro, además de los Caracteres "CR" y "LF". Para que quede claro el final del archivo, hay un registro de fin de archivo, el cual no contiene elementos solo los bytes de encabezado conteniendo caracteres "0" ASCII.

Ejemplo de un archivo tipo hex:

Programa ensamblado:

```
                ORG 1435H
1435 3E 23      LD  A,23H
1437 21 34 12   LD  HL,1234
                ORG  2000H
2000 07 07 07   DB  7,7,7
2003 07         DB  7
```

Archivo HEX:

```
:05143500
:042000000707070700
:00000001FF
```

En el ejemplo anterior el primer reg. tiene cinco

elementos elementos: 3E, 23, 21, 34 y 12. además se ensambló apartir de la localidad 1435hex. Vemos que el archivo solo contiene dos registros ademas del reg. de fin de archivo.

Rutinas de la Biblioteca del Macroensamblador.

Ya que para la transferencia del archivo se utilizaron rutinas externas de la biblioteca del Macro haremos mención de estas.

Rutinas de conversiòn:

ADEC: Conversiòn Decimal a Binario. Esta convierte una cadena de caracteres ASCII correspondiente a un número BCD a un valor Binario en el par de reg.HL. por otra parte los reg.BC deben apuntar a la cadena antes de hacer la llamada.

AHEX:Conversiòn ASCII Hex. a binario. Esta es similar a la anterior y convierte una cadena de caracteres ASCII correspondiente a un valor Hexadecimal a un valor binario en el par de reg.HL.(al final de la cadena deberà estar el caracter ASCII "H") Los reg.BC se utilizan en igual forma que la anterior llamada.

Rutinas para el Manejo de Archivos.

FNAME: Esta rutina construye un Bloque de control de archivo Extra. Si esta rutina es llamada con el reg. A igual a cero, la extensiòn (tipo) del archivo es utilizada y para el caso contrario que el reg A no sea igual a cero, los registros A,B y C contendràn la extensiòn del archivo. En el programa de transferencia los registros: A,B y C. Se cargaron con los caracteres 'H','E' Y 'X' respectivamente ya que el archivo que se va a utilizar siempre debe de ser tipo HEX. En caso de tratar con un archivo que tenga otra extensiòn, omitirà la extensiòn y buscara un archivo con el mismo nombre pero tipo HEX.

ZOPN: Esta rutina abre un archivo ya existente. En caso de ocurrir un error en la operaciòn se "prende" la bandera de CERO (Z).

ejemplo:

```
LD   DE,BCAEXT   ;apuntador al BCA extra.
CALL ZOPN        ;apertura de archivo.
CALL Z,ZIOER     ;rutina de error.
```

ZIOER: Esta es una rutina general de error. Cuando ocurre un error con algunas de las rutinas que manejan archivos, los reg. HL se cargan como apuntadores de un mensaje de error que es desplegado y se regresa el control al sistema operativo CDOS.

ZCLOS: Mediante esta rutina podemos cerrar archivos y se utiliza en forma similar a ZOPN.

GCHAR: Esta rutina extrae un caracter ASCII de un disco o de algùn dispositivo (consola, lectora, etc.). Cuando un registro "random" sin escribir es leido se toma como fin de archivo.

ejemplo:

```
        LD      DE,BCAEXT   ;apuntador al BCA extra.
        CALL    GCHAR       ;extrae un caracter
        CP      1AH         ;verifica si
        JP      Z,EOF       ;fin de archivo.
```

PRNT:Mediante esta llamada podemos imprimir letreros. Para utilizarla basta cargar los reg. HL con la localidad donde empieza el mensaje, ademàs colocar un caracter "CR" o un caracter "0" al final de nuestro mensaje. En caso que el mensaje termine con un "CR", se enviarà un "LF" y un "CR".

ejemplo:

```
        LD    DE,BCAEXT
        LD    HL,MENSAJ
        CALL  PRNT

MENSAJ:  DB   'ESTO ES EL MENSAJE'
```

ABORT: Esta rutina aborta el programa hacia CDOS e imprime un mensaje. El formato para el mensaje es igual que el de la rutina anterior.

ejemplo:

```
        LD      HL,MENSAJ
        CALL    ABORT

MENSAJ: DB      'FIN DE TRANSFER.'
```

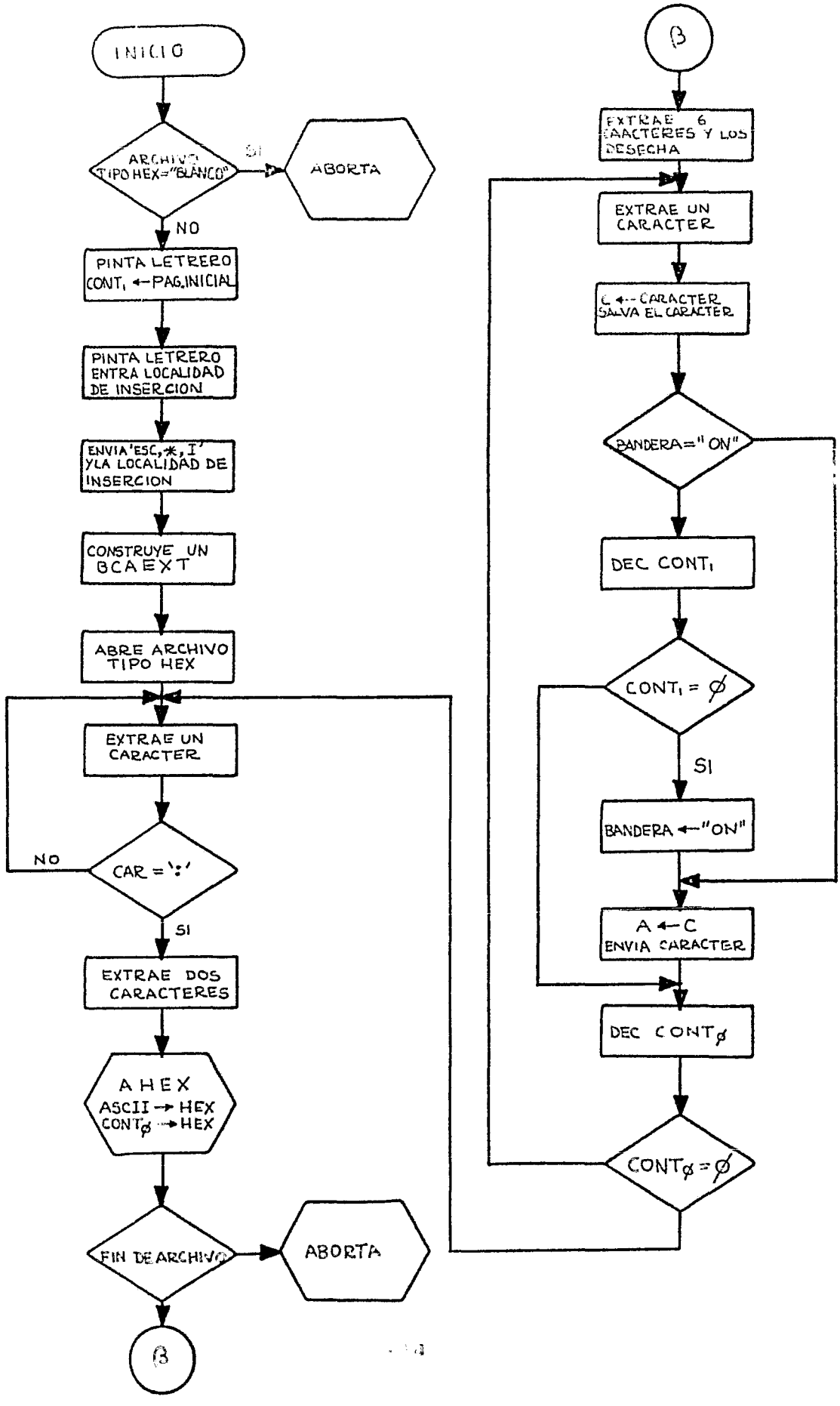
Además de las rutinas externas se utilizo una rutina externa que fué diseñada por nosotros.

BCDIN:Esta rutina de entrada es para números BCD. Cuando se llama a esta rutina permite solo la entrada de caracteres entre '0' y '9', para caracteres diferentes marca error haciendo sonar el timbre de la terminal. Esta rutina va colocando los caracteres ASCII que van siendo tecleados ('0'...'9') en un buffer de 32 localidades. Para meter los caracteres al buffer contamos con un apuntador el cual se incrementa cada vez que que entra un caracter, para poder controlar el apuntador se permite la entrada al caracter de retroceso (ctrlC) con el cual podemos retroceder el apuntador hasta la base del buffer. Mediante un contador controlamos que el apuntador no se salga de los limites del buffer. Al entrar un caracter "CR" se determina la entrada del último caracter BCD que queremos introducir. A partir de aquí solo se decrementa el apuntador cuatro veces y se hace una llamada a ADEC (BCD a binario).

Descripción general del Programa de Transferencia.

Como dijimos anteriormente el programa de Transferencia solo se encarga de abrir el archivo tipo hex. y vaciar su contenido hacia la terminal y el Starter Kit. Le pasamos primeramente el nombre del archivo tipo HEX., el numero de la pagina(256 Bytes) a partir de la cual se hara la transferencia (hay casos que no se desea transferir todo el programa o no se puede transferir por limitaciones en la memoria del Starter Kit) y la direcciòn desde la cual se hara la inserciòn.

El programa verifica si el nombre que damos es diferente de un espacio(no tenemos archivo para trabajar) y luego recibe los paràmetros mencionados anteriormente. En este momento envia la localidad inicial de inserciòn al Starter Kit, abre el archivo tipo HEX y toma como contador los dos bytes que siguen a caracter ":" dentro del archivo, pero ya que esto bytes estan en ASCII tiene que convertirlos a binario, lo cual realiza mediante la llamada AHEX. Con un contador(pagina inicial) se determina si el contenido extraido del archivo se desecha o se envia a la terminal y con el otro contador se extrae el contenido de un registro del archivo en su totalidad.

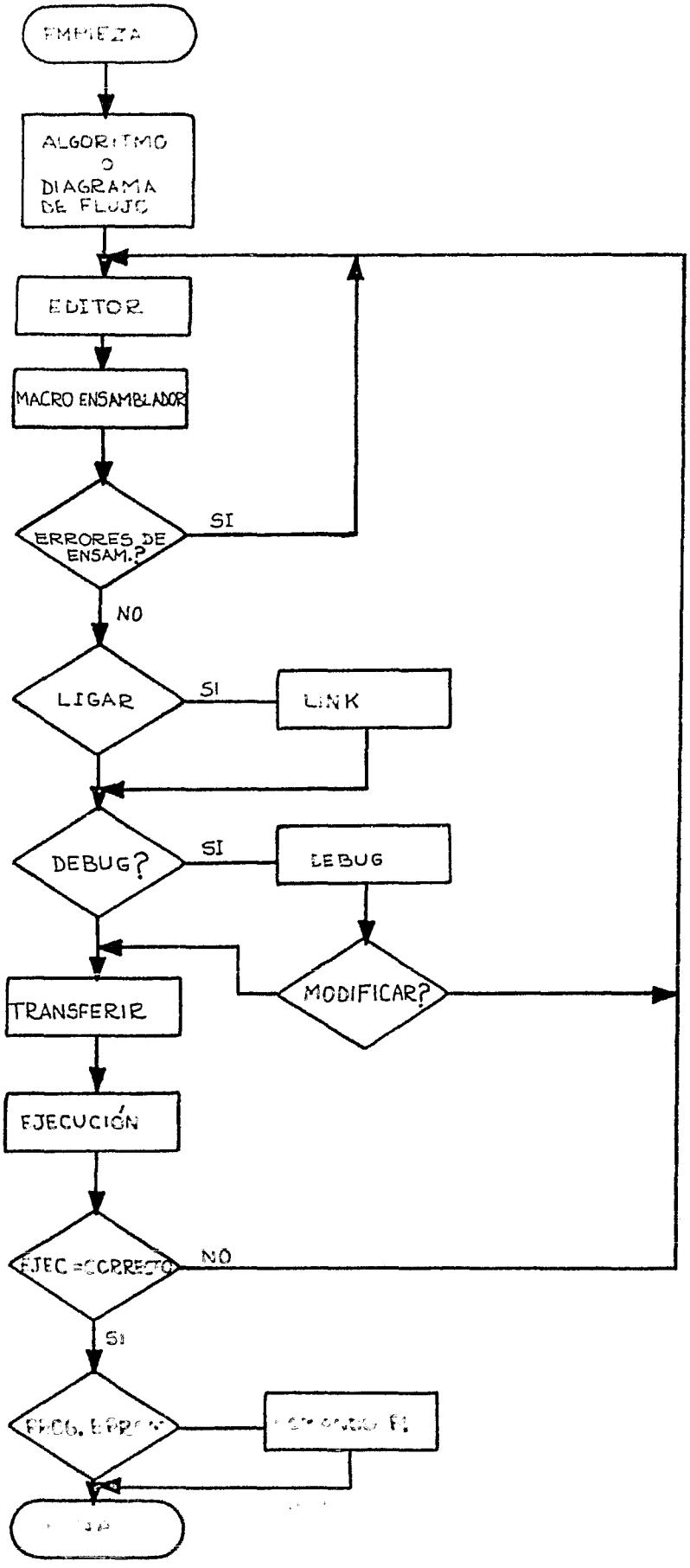

INICIO
ARCHIVO TIPO HEX="BLANCO"
SI
ABORTA
NO
PINTA LETRERO CONT1 ← PAG.INICIAL
PINTA LETRERO ENTRA LOCALIDAD DE INSERCION
ENVIA 'ESC,*,I' Y LA LOCALIDAD DE INSERCION
CONSTRUYE UN BCAEXT
ABRE ARCHIVO TIPO HEX
EXTRAE UN CARACTER
NO
CAR = ':'
SI
EXTRAE DOS CARACTERES
A HEX
ASCII → HEX
CONTØ → HEX
FIN DE ARCHIVO
ABORTA
B
B
EXTRAE 6 CARACTERES Y LOS DESECHA
EXTRAE UN CARACTER
C ← CARACTER SALVA EL CARACTER
BANDERA = "ON"
DEC CONT1
CONT1 = Ø
SI
BANDERA ← "ON"
A ← C
ENVIA CARACTER
DEC CONTØ
CONTØ = Ø

Desarrollo de un Programa

en el Sistema de Desarrollo por Variado.

El primer paso en el desarrollo de un programa es describir el programa mediante un diagrama de flujo o con el algoritmo. Después de ésto se crea el programa fuente mediante un editor de textos y se procede a ensamblarlo (utilizando la opciòn: HEX), en caso de no tener ningùn error en la sintaxis se pude continuar y en caso contrario deberemos regresar al editor para hacer correcciones. Para continuar se pueden probar partes del programa con el programa Debugger o transferir nuestro programa directamente al Starter Kit. En algunos casos se puede ensamblar el programa sin la opciòn Hex, esto con el objetivo de poder ligarlo a otro programa o alguna biblioteca en éste caso se utilizarà la opciòn "X"(equivalente a la HEX del ensamblador) del LIGADOR, con lo cual crearemos un archivo tipo HEX. que contendrà el programa en desarrollo y el programa o rutinas ligadas. Como se explico en la parte referente al programa de Transferencia se deberà pasar como paràmetros: el archivo a transferir (debera ser extensiòn HEX), el numero de pagina desde la cual se trasferira el archivo y la localidad de memoria en el Starter Kit desde donde se insertarà el programa. Una vez realizada la transferencia el control se deberà pasar al Programa Monitor con el cual podremos hacer modificaciones en el programa, probarlo y si deseamos

salvarlo en una memoria EPROM. En caso de querer hacer una modificaiòn grande en nuestro programa podemos volver al paso de ediciòn y repetir todos los pasos hasta lograr que nuestro programa cumpla los objetivos deseados.


EMPIEZA
ALGORITMO O DIAGRAMA DE FLUJO
EDITOR
MACRO ENSAMBLADOR
ERRORES DE ENSAM.?
SI
NO
LIGAR
SI
LINK
DEBUG?
SI
DEBUG
MODIFICAR?
TRANSFERIR
EJECUCIÓN
NO
SI

Ejemplo del Desarrollo de un Programa.

Para ejemplificar el anterior diagrama de flujo desarrollaremos un programa para un "Reloj de tiempo Real". Este programa en tiempo real representa la rutina basica de un programa de control. Nuestro programa es basicamente una rutina de interrupciòn que actualiza un contador ("Reloj"), pero para que pueda ocurrir la interrupciòn debemos hacer que se inicialicen ciertos paràmetros que permiten la interrupciòn.

La interrupciòn en este caso es producida cada segundo por el CTC el cual se programa como contador y recibé una señal externa de 60Hz. derivada de la linea de C.A.

Los pasos a seguir para el desarrollo del programa son:

Paso 1.-Algoritmo o Diagrama de flujo.

En la siguiente pagina se muestra el diagrama de nuestro programa: "Reloj de Tiempo Real".

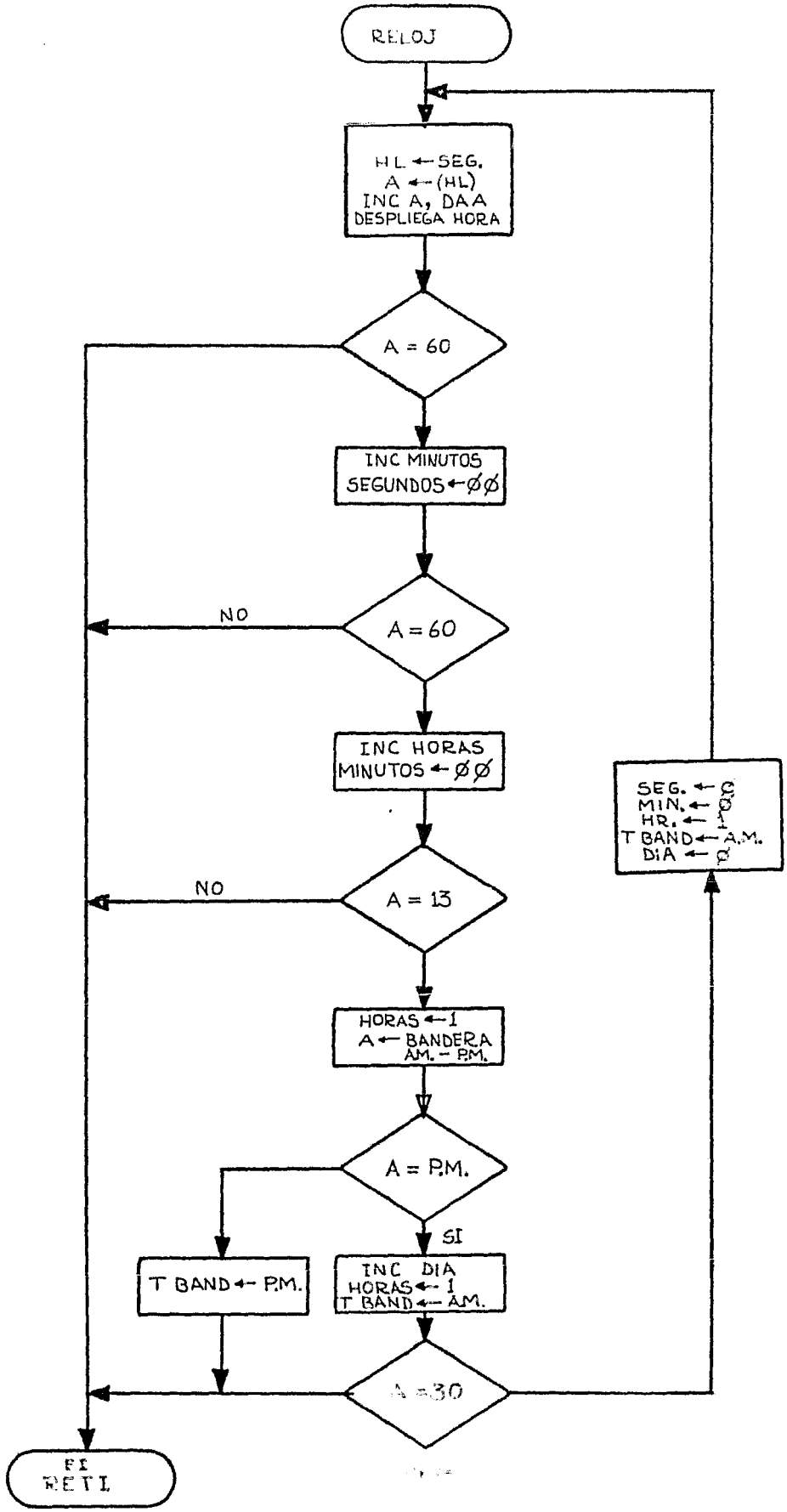
Paso 2.-Ediciòn del Programa.

Estando bajo el control de CDOS podemos invocar un editor de texto bajo cualquier nombre y extensiòn Z80.

ejem:

A.Screen RELOJ.Z80

En el ejemplo anterior "A." es un indicador que estamos bajo el control de CDOS y que el Disco que utiliza el sistema por default es el "A" (tenemos dos ops: A y B) "Screen" es un editor de Textos de Tromenco y nuestro programa lo hemos denominado "RELOJ" la extensiòn "Z80" indica que es un


RELOJ
HL ← SEG.
A ← (HL)
INC A, DAA
DESPLIEGA HORA
A = 60
NO
INC MINUTOS
SEGUNDOS ← ØØ
A = 60
NO
INC HORAS
MINUTOS ← ØØ
A = 13
NO
HORAS ← 1
A ← BANDERA
AM. - P.M.
A = P.M.
SI
T BAND ← P.M.
INC DIA
HORAS ← 1
T BAND ← A.M.
A = 30
SEG. ← Ø
MIN. ← Ø
HR. ← 1
T BAND ← A.M.
DIA ← Ø
EI
RETI

programa fuente para el Macroensamblador.

Paso 3.-Ensamblado.

Una vez que salimos del editor aparece en la terminal el mensaje:

A. n k escritos en "RELOJ.Z80". Este mensaje nos indica que nuestro programa fuente tiene un tamaño de "n" K bytes.

Ahora podemos proceder a ensamblar el programa invocando al Macroensamblador:

A."ASMB RELOJ.ØØØ HEX"

En la expresiòn anterior "A." es el indicador de CDOS y lo que aparece entre comillas lo escribimos para invocar al macroensamblador. ASMB el programa Macroensamblador RELOJ el programa fuente y la extensiòn "ØØØ" indica que el programa fuente tiene extensiòn Z80, que se encuentra en el disco que està utilizando el sistema por default y que deseamos un listado del ensamblado "RELOJ.PRN". Por ultimo "HEX" indica que queremos una archivo tipo hex como salida del Macroensamblador.

El ensamblador indico los siguientes errores:

```
LD      HL,TIME
"undefinided symbol at line 110"
LD      A,(TBAND)
"undefinided symbol at line 145"
```

En el primer intento de ensamblado tuvimos dos errores por lo que tuvimos que regresar al paso2 para modificar el programa fuente. En el segundo intento no tuvimos errores por

lo que podemos seguir adelante.

Paso 4.1-LIGADOR

Este paso es opcional y se utiliza cuando se desea ligar rutinas al programa en desarrollo, para ésto invocamos al programa ligador de la siguiente manera:

A."LINK RUT,RELOJ,RELOJ1/N/X/E"

Esta expresiòn es similar a la del ensamblador. "A." es el indicador de CDOS, LINK es el programa ligador, RUT es el archivo ensamblado donde se encuentran la rutinas del Programa Monitor, RELOJ es el Programa en desarrollo ya ensamblado al cual se le ligaran las rutinas de RUT, RELOJ1 es el nombre de nuestro archivo de salida (que serà de tipo HEX), las opciones /N/E hacen que el ligador genere un nuevo archivo de los que se le pasaron como paràmetros y la opción /X hace que éste nuevo archivo sea de tipo HEX.

Paso 4.2- Debugger

Este paso es opcional y sirve para probar algunas rutinas en el sistema Cromenco. para ésto corremos el programa Debugger de la siguiente manera:

A.DEBUG RELOJ.HEX

Con el Debug se puede monitorear el programa dentro del sistema Cromenco y si encontramos algunos errores podemos regresar al paso 2.

Paso 4.3- Transferencia

En éste paso corremos el programa de transferencia de la siguente manera:

A.TRANS RELOJ

Donde "TRANS" es nuestro programa de transferencia hacia el Starter Kit y RELOJ es el programa a tranferir en este caso. Si no hay un archivo que se llame RELOJ.HEX el programa marcara error y abortara. Cuando el programa TRANS empieza a correr despliega lo siguiente:

```
**Se borra la pantalla de la terminal**
Programa de transferencia
teclee la Pagina (256 bytes)
a partir de la cual
se hara la transferencia:"0"
Teclee la localidad de inserciòn en Hexa. XXXX
se tomaran los ultimos cuatro digitos:"2000"
```

Aquì se observa como se pasaron los paràmetros al programa. Cuando estamos haciendo la transferencia tenemos la opciòn de abortar ésta con la tecla "Escape" o con CtrlC.

Paso 5.1-Ejecuciòn

En este paso el control pasa al programa Monitor y aquì hacemos las pruebas finales del programa, si queremos hacer pequeños cambios lo podemos hacer aquì de otra manera tenemos que regresar al paso 2.

Paso 5.2.-Programaciòn de EPROM.

Este paso es opcional y se realiza cuando el proyecto esta totalmente depurado.

Bibliografia.

Auiman M. ;El Empleo de los Microprocesadores
Ed:Toray Masson, 1981, p.p.208.

Auslander David M.,
Sagues Paul ;Microprocessors for
Meaeurament and Control
Ed:Osborne/Macgraw-Hill, 1981, p.p. 310.

Bennett William S.,
Everet Jr. Carl f;What every engineer Should Know
About Microcomputers
Ed:Marcel Dekker Inc., 1980, p.p.175

Bibbero Robert,
Stern David ;Microprocessors Systems,
Interfacing and Aplications
Ed:John Wiley & Sons. , 1982, p.p.195

Ciarcia Steve ;Build Your Own Z80 Computer
Ed:Byte Books/Macgraw-Hill, 1981, p.p.332.

Cromenco ;Z80 Macro Assembler inst. Manual.
Ed:Cromenco, 1982,p.p.206.

Garetz Mark,
Libes Sol ;Interfacing to S-100/IEEE 696
Microcomputers.
Ed:Osborne/Macgraw-Hill, 198, p.p.310

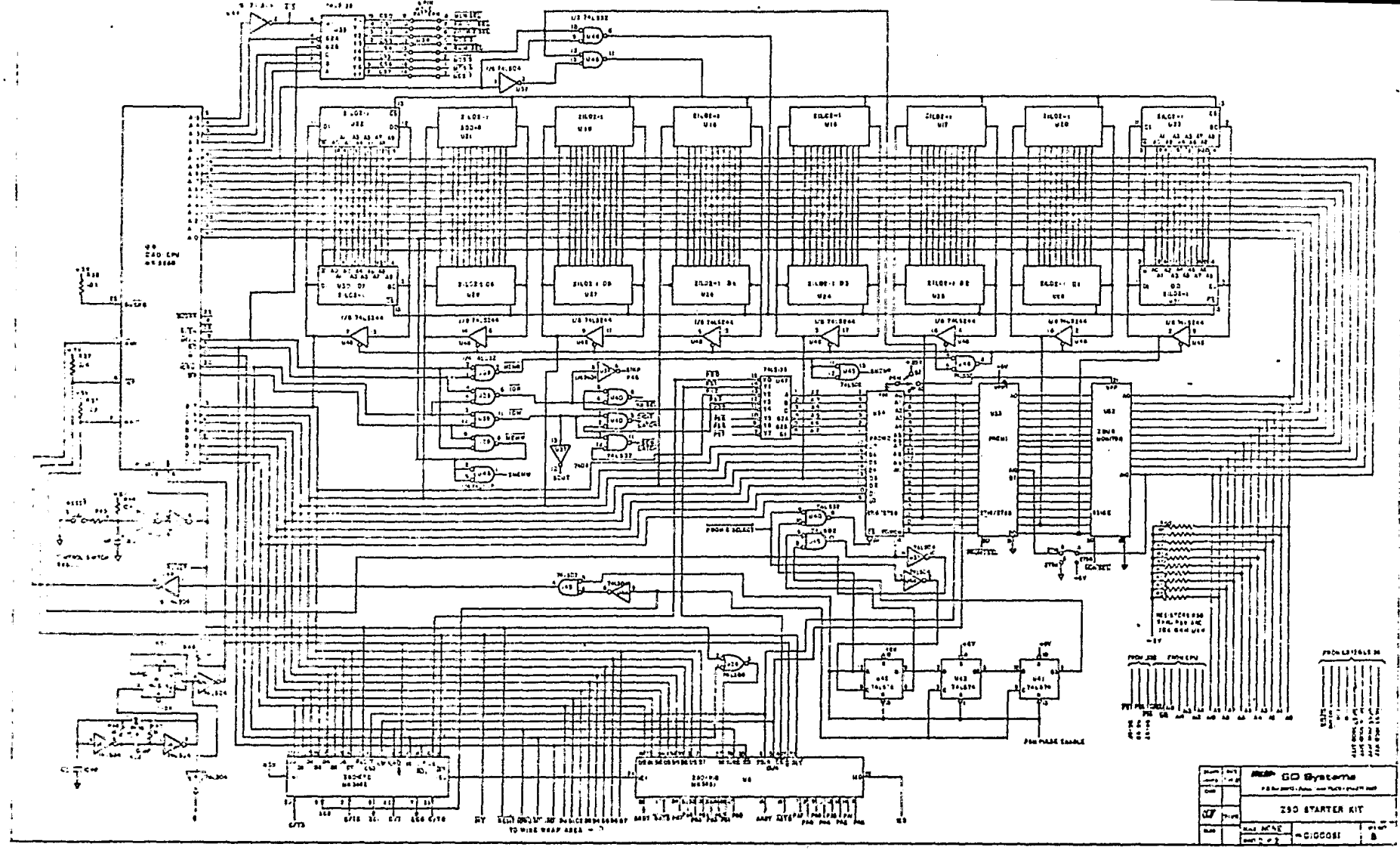
Miller Alark ;The 8080/Z80 Assembly Language
Techniques for improved Programming
Ed:Jhon Wiley & Sons, 1981, p.p. 317

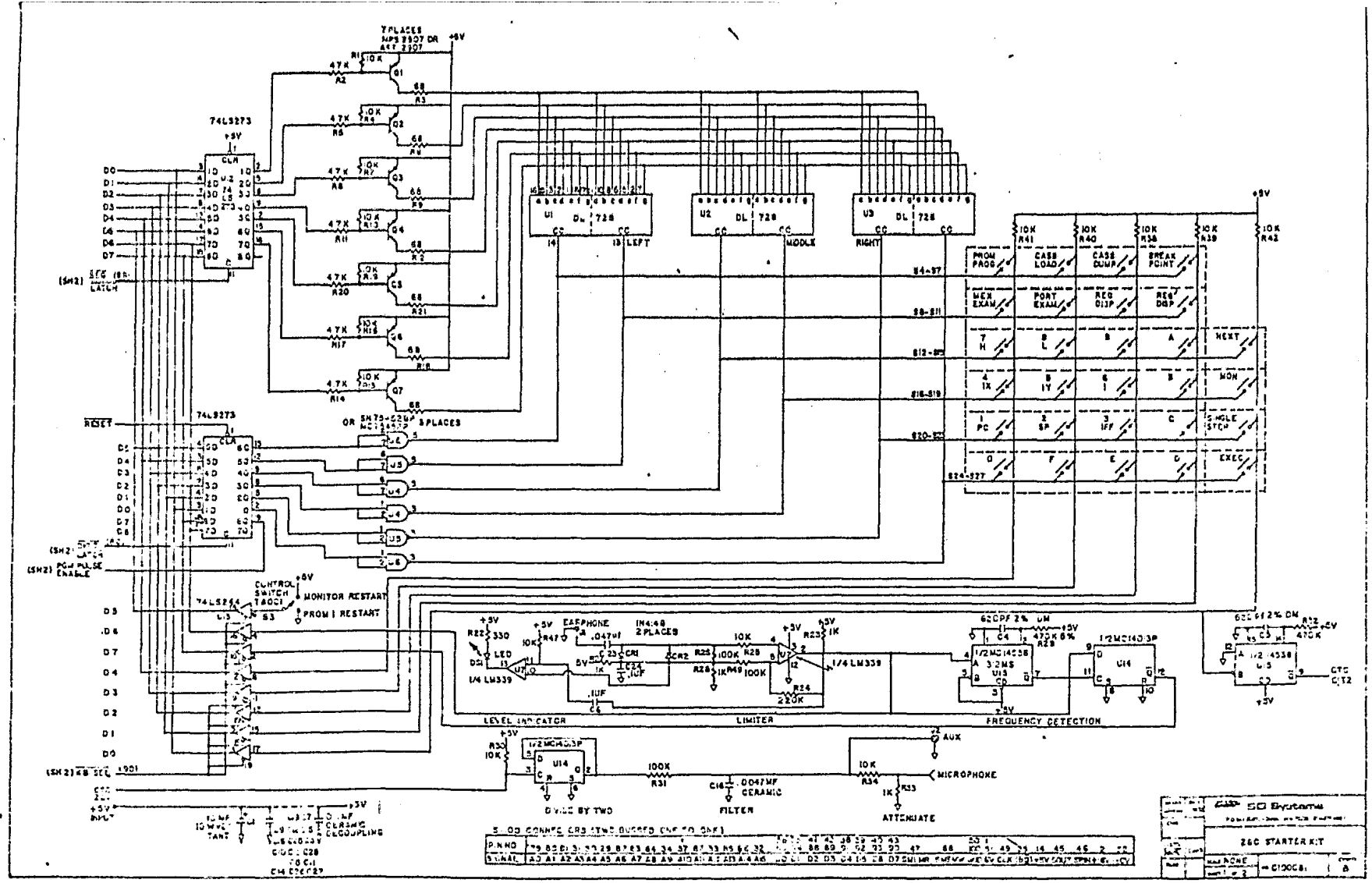
Peatman Jhon B.;Microcomputers Based Desing
Ed:Macgraw-Hill/Kogakusha., 1977, p.p. 586.

Sale Rodnay ;How to Program the Z80
Ed:Sybex, 1980, p.p. 224.

APENDICE I.

Diagrama del Equipo Estarter Kit Z80.


GO Systems
Z80 STARTER KIT
SIZE
B


74LS273
74LS244
LEVEL INDICATOR
LIMITER
FREQUENCY DETECTION
DIVIDE BY TWO
FILTER
ATTENUATE
MICROPHONE
AUX
LEFT
MIDDLE
RIGHT
MONITOR RESTART
PROM RESTART
EARPHONE
PROM PROG
CASS LOAD
CASS DUMP
BREAK POINT
MEM EXAM
PORT EXAM
REG DISP
NEXT
MON
SINGLE STEP
EXEC
SBC 2&C STARTER KIT

APENDICE II.

Listado de las Rutinas del Programa Monitor.

```
<* RUTINAS DEL PROGRAMA MONITOR*>

*********************************
*                               *
*     RUTINAS DEL PROGRAMA      *
*           MONITOR             *
*********************************
                    0007
                    0008  ; VARIABLES GLOBALES:
                    0009          GLOBAL  BAND,BYTTY,CO16,CONV
                    0010          GLOBAL  CRECO,CTRLC,ECO,PINTA
                    0011          GLOBAL  POLENT,RESTA,SALHL,SALTTY
                    0012          GLOBAL  ERROR,STACK,RECCOM,PASCII
                    0013
      (0007)        0014  CTRL6   EQU     07H
      (0020)        0015  ESP     EQU     20H
      (000D)        0016  CR      EQU     13
      (000A)        0017  LF      EQU     10
      (0098)        0018  USART   EQU     98H
      (003E)        0019  PROMPT  EQU     '>'
      (0000)        0020  KIT     EQU     00
      (0001)        0021  CRO     EQU     01
                    0022  *INCLUDE B:EQUIPO.Z80
      (0000)        0023  ENSAM   EQU     KIT
                    0024
                          (***** end of include *****)
                    0025  CORRI   MACRO
                    0026          ADD     HL,HL
                    0027          ADD     HL,HL
                    0028          ADD     HL,HL
                    0029          ADD     HL,HL
                    0030          MEND
                    0031  ESPAC   MACRO
                    0032          LD      A,ESP
                    0033          CALL    SALTTY
                    0034          MEND
                    0035          REL
                    0036  ;**********************************
                    0037  ; RUTINA DE CONVERSION DE ASCII A HEXA.
                    0038  ;**********************************
                    0039  CONV
0000' E5            0040          PUSH    HL              ; Salva reg.
0001' 210F00'       0041          LD      HL,CONVTAB+0FH  ; Carga apuntador a tabla.
0004' 011000        0042          LD      BC,10H          ; contador de long de la tabla
0007' EDB9          0043          CPDR                    ; Se compara w Z=1 si el valor
0009' E1            0044          POP     HL              ; esta en la tabla con lo que
000A' C8            0045          RET     Z               ; restauran reg. y se regresa
000B' C34C00'    R  0046          JP      ERROR           ; Si la bandera Z=0 se va al
                    0047                                  ; Coman. de Error ya que el valor
                    0048                                  ; esta fuera de la tabla.
                    0049  ;**********************************
                    0050  ; RUTINA DE CONV DE HEX A ASCII
                    0051  ;**********************************
                    0052  HEXCO
000E' C5            0053          PUSH    BC              ; Salvar reg.
000F' 0E0A          0054          LD      C,0AH
0011' B9            0055          CP      C               ; Dato <10 (dec.)
0012' FA1900'       0056          JP      M,SUM30         ; si es asi sumar 30hex.
0015' C637          0057          ADD     37H             ; en otro caso sumar 37hex.
0017' C1            0058          POP     BC              ; Restaurar reg y regresar.
0018' C9            0059          RET
0019' C630          0060  SUM30   ADD     30H             ; Dato=Dato+30h
001B' C1            0061          POP     BC              ; Restaurar reg w regresar.
001C' C9            0062          RET
                    0063  ;**********************************
                    0064  ; CONV UN BYTE  EN 2 CARACTERES ASCII
                    0065  ; Y LOS ENVIA A TTY.
                    0066  ;**********************************
                    0067  BYTTY
001D' E5            0068          PUSH    HL              ; Salvar registros.
001E' D5            0069          PUSH    DE
001F' C5            0070          PUSH    BC
0020' 4F            0071          LD      C,A             ; Salvar dato.
0021' CB3F          0072          SRL     A               ; Se corre el nibble
0023' CB3F          0073          SRL     A               ; mas significativo
```

```
0025' CB3F      0074          SRL     A               ; del Dato hacia el
0027' CB3F      0075          SRL     A               ; el nibble menos sig.
0029' CD0E00'   0076          CALL    hexco           ; Se convierte a ASCII
002C' CDAF00'   0077          CALL    SALTTY          ; y se evia a la terminal.
002F' 79        0078          LD      A,C             ; se restaura el dato original
0030' E60F      0079          AND     0FH             ; se mascara el nibble mas sig.
0032' CD0E00'   0080          CALL    hexco           ; Se convierte a ASCII
0035' CDAF00'   0081          CALL    SALTTY          ; y se envia a la terminal
0038' C1        0082          POP     BC              ; Restaura los regs.
0039' D1        0083          POP     DE
003A' E1        0084          POP     HL
003B' C9        0085          RET
                0086  ;*************************************
                0087  ; RESTA DE 16 BITS HEX
                0088  ;MINUENDO EN HL
                0089  ;SUSTRAENDO EN DE
                0090  ;*************************************
                0091  RESTA
003C' 37        0092          SCF                     ; CY=1
003D' 3F        0093          CCF                     ; CY=CY'=0
003E' ED52      0094          SBC     HL,DE           ; HL=HL+DE
0040' DA4C00'   R 0095        JP      C,ERROR         ; SI HL<0 ERROR
0043' 221100'   0096          LD      (LONG),HL       ; Salva el resultado
0046' ED531000' 0097          LD      (BASE),DE       ; Salva el operando menor.
004A' 23        0098          INC     HL              ; Inc. la diferencia
004B' C9        0099          RET
                0100  ;*************************************
                0101  ;  Rutina de Error.
                0102  ;*************************************
004C' 3E07      0103  ERROR   LD      A,CTRLG         ; Envia un CtrlG ASCII
004E' CDAF00'   0104          CALL    SALTTY          ; (campana).
0051' 3E2A      0105          LD      A,'*'           ; Envia un asterisco como
0053' CDAF00'   0106          CALL    SALTTY          ; simbolo de que ocurrio un error
0056' CDD800'   0107          CALL    CRECO           ; Salta a una nueva linea.
0059' 310000#   0108          LD      SP,STACK        ; Reinicializa el SP.
005C' C30000#   0109          JP      RECCOM          ; Regresa al reconocimiento
                0110                                  ; de comandos.
                0111  ;*************************************
                0112  ;RUTINA QUE CONVIERTE 4 CARACTERES ASCII
                0113  ; A UN ENTERO 16 BITS
                0114  ;*************************************
                0115  CO16
005F' C5        0116          PUSH    BC              ; Salvar registros.
0060' D5        0117          PUSH    DE
0061' 210000    0118          LD      HL,00           ; HL=0000.
0064' CDA400'   0119          CALL    POLENT          ; Entra un dato
0067' CDC500'   0120          CALL    ECO             ; el Eco del dato se envia a term.
006A' CD0000'   0121          CALL    CONV            ; Se convierte el dato a Hex.
006D' 79        0122          LD      A,C             ; El dato se coloca en el
006E' B5        0123          OR      L               ; nibble menos sig. del reg. L
006F' 6F        0124          LD      L,A
0070' CDA400'   0125  ECO16   CALL    POLENT          ; Entra un dato
0073' CDC500'   0126          CALL    ECO             ; el ECo del dato se envia a term.
0076' FE20      0127          CP      ESP             ; Si el dato que entro= "espacio",
0078' 2818      0128          JR      Z,POP16         ; "coma","dos puntos" o "CR"
007A' FE0D      0129          CP      CR              ; se termina la rutina.
007C' 2816      0130          JR      Z,POP16
007E' FE2C      0131          CP      ','
0080' 2810      0132          JR      Z,POP16
0082' FE3A      0133          CP      ':'
0084' 280C      0134          JR      Z,POP16
0086' CD0000'   0135          CALL    CONV            ; Se convierte el dato a Hex.
0089'           0136          CORRI                   ; HL se corre cuatro veces a la iz.
0089' 29        0137+         ADD     HL,HL
008A' 29        0138+         ADD     HL,HL
008B' 29        0139+         ADD     HL,HL
008C' 29        0140+         ADD     HL,HL
008D' 79        0141          LD      A,C             ; el dato se coloca en el
008E' B5        0142          OR      L               ; nibble menos sig. del reg. L
008F' 6F        0143          LD      L,A
0090' 18DE      0144          JR      ECO16           ; Se repite la operacion hasta
                0145                                  ; que entre uno de los cuatro teres de
                0146                                  ; Control mencionados.
```

```
(* RUTINAS DEL PROGRAMA MONITOR*)

0092' D1          0147  POP16:  POP     DE              ; Se restauran los reg.
0093' C1          0148          POP     BC
0094' C9          0149          RET
                  0150  ; SIGUE UN BLOQUE DE ENSAMBLADO CONDICIONAL
      (0000)      0151          IF ENSAM EQ CRO         ; Sigue bloque ensamblado para
                  0152                                  ; correr en Cromenco.
                  0153
                  0154  ;*********************************
                  0155  ;*  Rutina de Escape y Ctrl C    *
                  0156  ;*********************************
                  0157  CtrlC:
                  0158          PUSH    AF              ; Salvar registros.
                  0159          PUSH    BC
                  0160          LD      C,0BH           ; C=11dec(rutina del status del teclado)
                  0161          CALL    05              ; revisa el status del teclado
                  0162          CP      00              ; Si no llego ningun dato
                  0163          JR      Z,CTRLF         ; regresa, y si no lee
                  0164          CALL    POLENT          ; el dato que llego.
                  0165          CP      1BH             ; Si el dato que llego es:
                  0166          JP      Z,ERROR         ; "escape" o "Ctrl-C" aborta.
                  0167          CP      03              ; Si no es uno de los caracteres
                  0168          JP      Z,ERROR         ; anteriores se desecha,
                  0169  CtrlF:  POP     BC              ; Restaura regs. y regresa.
                  0170          POP     AF
                  0171          RET
                  0172  ;*************************************
                  0173  ; RUTINA DE SALIDA A TTY
                  0174  ;*************************************
                  0175  SALTTY
                  0176          PUSH    BC              ; Salvar registros
                  0177          PUSH    DE              ; Cargo el valor a enviar
                  0178          LD      E,A             ; a la terminal.
                  0179          LD      C,02            ; C=2 (rutina de salida a terminal)
                  0180          CALL    05              ; Envia el caracter a la terminal.
                  0181          POP     DE              ; Restaura los registros.
                  0182          POP     BC
                  0183          RET
                  0184  ;*************************************
                  0185  ;RUTINA DE ENTRADA POR POLLING
                  0186  ;*************************************
                  0187  POLENT
                  0188          PUSH    BC              ; Salvar registros
                  0189          PUSH    DE              ; C=00h (rutina de entrada
                  0190          LD      C,B0H           ; sin Eco).
                  0191          CALL    05              ; Entra el valor ASCII en
                  0192          POP     DE              ; el reg.  A.
                  0193          POP     BC              ; Restaura los registros.
                  0194          RET
                  0195          ELSE
                  0196  ;*********************************
                  0197  ;*   Rutina de control-C y Escape *
                  0198  ;*********************************
                  0199  CtrlC
0095' 08          0200          EX      AF,AF'          ; Salvar el reg. A
0096' DB98        0201          IN      A,(USART)       ; leer ultimo dato del Usart
0098' FE1B        0202          CP      1BH             ; Dato=escape?
009A' CA4C00'   R 0203          JP      Z,ERR)R         ; Si es igual Aborta.
009D' FE03        0204          CP      03              ; Dato=Ctrl-C?
009F' CA4C00'   R 0205          JP      Z,ERROR         ; Si es igual Aborta.
00A2' 08          0206          EX      AF,AF           ; Restaura A.
00A3' C9          0207          RET
                  0208  ;*********************************
                  0209  ;* Rutina de entrada por polling *
                  0210  ;*********************************
                  0211  Polent:
00A4' DB99        0212          IN      A,(USART+1)     ; ver status
00A6' CB4F        0213          BIT     1,A             ; Status=Listo?
00A8' 28FA        0214          JR      Z,POLENT        ; Si no esta listo regresa
                  0215                                  ; a leer el status.
00AA' DB98        0216          IN      A,(USART)       ; Entra dato
00AC' E67F        0217          AND     7FH             ; Mascarar bit 7
00AE' C9          0218          RET
                  0219
```

```
                        0220  ;*********************************
                        0221  ;* Rutina de salida a terminal  *
                        0222  ;*********************************
                        0223  Saltty:
00AF' 08                0224          EX      AF,AF'          ; salvar dato
0080' DB99              0225  Status: IN      A,(USART+1)     ; ver status
0082' CB47              0226          BIT     0,A             ; Si status= no esta lista la
0084' 28FA              0227          JR      Z,STATUS        ; salida regresa a ver status.
0086' 08                0228          EX      AF,AF'          ; Status=listo, restaura dato
0087' D398              0229          OUT     (USART),A       ; sale dato
0089' C9                0230          RET
                        0231          ENDIF                   ; Fin de ensamblado cond.
                        0232                                  ; si ensam=kit
                        0233  ;*********************************
                        0234  ;SALIDA DE HL
                        0235  ;*********************************
                        0236  SALHL
008A' F5                0237          PUSH    AF              ; Salva registros.
008B' 7C                0238          LD      A,H             ; Envia contenido Hex.
008C' CD1D00'           0239          CALL    BYTTY           ; del reg. H a la term.
008F' 7D                0240          LD      A,L             ; Envia contenido Hex.
0090' CD1D00'           0241          CALL    BYTTY           ; del reg. L a la term.
0093' F1                0242          POP     AF              ; Restaura regs.
0094' C9                0243          RET
                        0244
                        0245  ;*********************************
                        0246  ;*         RUTINA DE ECO         *
                        0247  ;*********************************
                        0248  ECO:
0095' F5                0249          PUSH    AF              ; Salva registros.
0096' FE0D              0250          CP      CR              ; Si el dato es un "CR"
0098' 2809              0251          JR      Z,FECO          ; llama a la rutina CRECO
009A' FE20              0252          CP      20H             ; Si el dato es caracter de
009C' 3808              0253          JR      C,FECO1         ; control regresa o de lo
009E' CDAF00'           0254          CALL    SALTTY          ; contrario envia dato a term.
00A1' 1803              0255          JR      FECO1           ; regresa.
00A3' CDD800'           0256  FECO:   CALL    CRECO
00A6' F1                0257  FECO1:  POP     AF              ; Restaura regs.
00A7' C9                0258          RET
                        0259  ;*********************************
                        0260  ;*    Rutina de eco del CR       *
                        0261  ;*********************************
                        0262  Creco:
00D8' 3E0D              0263          LD      A,CR            ; Se envia un "CR",
00DA' CDAF00'           0264          CALL    SALTTY          ; un "LF" y el caracter
00DD' 3E0A              0265          LD      A,LF            ; "-" a la terminal.
00DF' CDAF00'           0266          CALL    SALTTY
00E2' 3E3E              0267          LD      A,PROMPT
00E4' CDAF00'           0268          CALL    SALTTY
00E7' 3E0D              0269          LD      A,CR            ; Se restaura A
00E9' C9                0270          RET
                        0271  ;*********************************
                        0272  ;*       Rutina Pinta          *
                        0273  ;*********************************
                        0274  PINTA:
00EA' C5                0275          PUSH    BC              ; Salva registros.
00EB' 0603              0276          LD      B,3             ; Contador=3
00ED' 1A                0277  PIN1    LD      A,(DE)          ; carga a con el contenido
00EE' CDAF00'           0278          CALL    SALTTY          ; de la tabla y se envia a
00F1' 13                0279          INC     DE              ; la terminal se incrementa
00F2' 10F9              0280          DJNZ    PIN1            ; apuntador. Se repite tres veces.
00F4' 3E3D              0281          LD      A,'='           ; Envia el caracter "=" a la terminal
00F6' CDAF00'           0282          CALL    SALTTY
00F9' C1                0283          POP     BC              ; Restaura los registros.
00FA' C9                0284          RET                     ; Fin de Pinta.
                        0285  ;*********************************
                        0286  ; RUTINA PARA DESPLEGAR BANDERAS
                        0287  ;*********************************
                        0288  BAND:
00FB' C5                0289          PUSH    BC              ; Salva los registros.
00FC' E5                0290          PUSH    HL
00FD' 0608              0291          LD      B,08            ; contador=8
00FF' 211200'           0292          LD      HL,FTAB         ; Apuntador a caracteres de
```

```
<* RUTINAS DEL PROGRAMA MONITOR*>

0102' 4F          0293          LD      C,A             ; banderas. Salva F en C
0103' CBE9        0294          SET     5,C             ; bits 5,3=1
0105' CBD9        0295          SET     3,C
0107' CB01        0296  Flag1:  RLC     C               ; Se verifica bit por bit
0109' 380A        0297          JR      C,ENVIA         ; si el bit=1 se envia el caracter
0108' 3E20        0298          LD      A,ESP           ; si no se envia un espacio
010D' CDAF00'     0299          CALL    SALTTY          ; a la terminal
0110' 23          0300          INC     HL              ; se incrementa el apuntador
0111' 10F4        0301          DJNZ    FLAG1           ; se repite hasta que  se revise
0113' 1807        0302          JR      FIN             ; cada bit de C.
0115' 7E          0303  Envia:  LD      A,(HL)          ; Se carga A con el caracter corresp.
0116' CDAF00'     0304          CALL    SALTTY          ; a la bandera y se envia a la terminal
0119' 23          0305          INC     HL              ; se incrementa el apuntador
011A' 10E8        0306          DJNZ    FLAG1           ; Se regresa a revisa      iguiente bit.
011C' E1          0307  FIN:    POP     HL              ; Restauran los regs.
011D' C1          0308          POP     BC
011E' C9          0309          RET                     ; Fin de Band.
                  0310  ;*************************************
                  0311  ;* RUTINA PARA PINTAR CARACTERES ASCII *
                  0312  ;*************************************
                  0313
011F' D9          0314  PASCII  EXX                     ; Salva los registros.
0120' E1          0315          POP     HL              ; HL=(SP)
0121' D1          0316          POP     DE              ; DE=M1
0122' E3          0317          EX      (SP),HL         ; (SP)=HL y HL=M2
0123' EB          0318          EX      DE,HL           ; HL=M1 Y DE=M2.
0124' B7          0319          OR      A               ; A=00
0125' F5          0320          PUSH    AF              ; Salva A
0126'             0321          ESPAC
0126' 3E20        0322+         LD      A,ESP
0128' CDAF00'     0323+         CALL    SALTTY
0128'             0324          ESPAC
0128' 3E20        0325+         LD      A,ESP
012D' CDAF00'     0326+         CALL    SALTTY
0130' ED52        0327          SBC     HL,DE           ; HL=M2-M1.
0132' 45          0328          LD      B,L             ; B=LONGITUD
0133' EB          0329          EX      DE,HL           ; HL=M1
0134' 7E          0330  PASC1:  LD      A,(HL)          ; Carga A con contenido del apunt.
0135' E67F        0331          AND     7FH             ; mascara el bit mas sig.
0137' 23          0332          INC     HL              ; incrementa apuntador
0138' FE20        0333          CP      20H             ; si es caracter de control
013A' 3002        0334          JR      NC,PASEN        ; se carga el A con un punto
013C' 3E2E        0335          LD      A,'.'           ; y se envia a la terminal
013E' CDAF00'     0336  PASEN:  CALL    SALTTY          ; en otro caso se envia el caracter.
0141' 10F1        0337          DJNZ    PASC1           ; se repite.
0143' D9          0338          EXX                     ; Se restauran registros.
0144' F1          0339          POP     AF
0145' C9          0340          RET                     ; Fin de Pascii.
                  0341
                  0342          DATA
                  0343  ;XXXXXXXX DATOS XXXXXXXXXXXXXXXXXXX
                  0344  ; Tabla de conversion de ASCII a Hex.
0000* 30313233    0345  CONVTAB: DB      '0123456789ABCDEF'
                  0346  ; Localidades para la rutina resta
0010* (0001)      0347  BASE    DS      1
0011* (0001)      0348  LONG    DS      1
                  0349  ; Tabla de caracteres de banderas para la rutina Band.
0012* 535A0048    0350  FTAB    DB      'SZ',0,'H',0,'VNC'
001A* (0000)      0351          END

Errors             0
Range Count        4

Program Length  0146 (326)
Data Length     001A (26)
```

<# RUTINAS DEL PROGRAMA MONITOR#>

Symbol		Value	Defn	References
BAND	E	00FB'	0009	#0288
BASE		0010'	0347	0097
BYTTY	E	001D'	0009	#0067 0239 0241
CO16	E	005F'	0009	#0115
CONV	E	0000'	0009	#0039 0121 0135
CONVTAB		0000'	0345	0041
CORRI		Macro	0025	0136
CR		000D	0016	0129 0250 0263 0269
CRECO	E	00D8'	0010	#0262 0107 0256
CRO		0001	0021	0151
CTRLC	E	0095'	0010	#0199
CTRLG		0007	0014	0103
ECO	E	00C5'	0010	#0248 0120 0126
ECO16		0070'	0125	0144
ENSAM		0000	0023	0151
ENVIA		0115'	0303	0297
ERROR	E	004C'	0012	#0103 0046 0095 0203 0205
ESP		0020	0015	0127 0298 0322 0325
ESPAC		Macro	0031	0321 0324
FECO		00D3'	0256	0251
FECO1		00D6'	0257	0253 0255
FIN		011C'	0307	0302
Flag1		0107'	0296	0301 0306
FTAB		0012'	0350	0292
HEXCO		000E'	0052	0076 0080
KIT		0000	0020	0023
LF		000A	0017	0265
LONG		0011'	0348	0096
PASC1		0134'	0338	0337
PASCII	E	011F'	0012	#0314
PASEN		013E'	0336	0334
PIN1		00ED'	0277	0280
PINTA	E	00EA'	0010	#0274
POLENT	E	00A4'	0011	#0211 0119 0125 0214
POP16		0092'	0147	0128 0130 0132 0134
PROMPT		003E	0019	0257
RECCOM	X	0000#	0012	0109
RESTA	E	003C'	0011	#0091
SALHL	E	008A'	0011	#0236
SALTTY	E	00AF'	0011	#0223 0077 0081 0104 0106 0254 0264 0266 0268 0278 0282 0297 0304 0323 0326 0336
STACK	X	0000#	0012	0108
Status		0080'	0225	0227
SUM30		0019'	0068	0056
USART		0098	0018	0201 0212 0216 0225 0229

APENDICE III.

Listado del Programa Monitor.

```
**********************************
*                                *
*                                *
*     MONITOR PARA TERMINAL      *
*      GERARDO NU&EZ PITTY       *
*         06/SEP/1983.           *
*                                *
*                                *
**********************************
Este programa monitor contiene diversos Comandos
los cuales son llamados desde una terminal que debera
estar conectada al equipo Starter Kit.
El programa Monitor puede correr en el sistema
Cromenco con limitaciones en algunos Comandos.
                0016
                0017  ; LIST se utiliza para listar el codigo de los DB y Macros.
                0018          LIST    GEN,TEXT
                0019  ; Variables globales:
                0020          GLOBAL  BAND,BYTTY,CO16,CONV
                0021          GLOBAL  CRECO,CTRLC,ECO,PINTA
                0022          GLOBAL  POLENT,RESTA,SALHL,SALTTY
                0023          GLOBAL  ERROR,STACK,RECCOM,FTAB
                0024          GLOBAL  ESP,LONG,PROMPT
                0025          GLOBAL  USART,PASCII
                0026
                0027
     (0000)     0028  NULL    EQU     00              ; Caracter nulo en ASCII.
     (0000)     0029  KIT     EQU     00H             ; Valores para los ensamblados condicionales
     (0001)     0030  CRO     EQU     01H             ; para el Satarter KIT,
     (0002)     0031  MC      EQU     02H             ; Sistema Cromenco y Mk-Z80.
     (003D)     0032  IGUAL   EQU     '='             ; Caracter ASCII: [=]
     (0020)     0033  ESP     EQU     20H             ; Caracter ASCII: [Espacio]
     (003E)     0034  PROMPT  EQU     '>'             ; Caracter ASCII: [>]
     (000D)     0035  CR      EQU     0DH             ; Caracter ASCII: [retorno de carro]
     (0007)     0036  CTRLG   EQU     07H             ; Caracter ASCII: [Campana]
     (000A)     0037  LF      EQU     0AH             ; Caracter ASCII: [salto de linea]
     (00D5)     0038  CTCPOS  EQU    0D5H             ; CTE DE CONTROL PARA INT CON FLANCO +
     (00C5)     0039  CTCNEG  EQU    0C5H             ;  '  '  '  '  '  '  '  '  '  '  '  -
     (0015)     0040  CTC16   EQU     15H             ; FACTOR DE ESCALA 16
     (0035)     0041  CTC256  EQU     35H             ; FACTOR DE ESCALA 256
                0042  *RELLIB RUT                     ; El archivo RUT.REL sera ligado a este programa
                0043  *INCLUDE B:EQUIPO.Z80           ; Se incluye un archivo que determina para
     (0000)     0044  ENSAM   EQU     KIT
                0045
                      (***** end of include *****)
                0046                                  ; que equipo sera el ensamblado y listado.
                0047  ;****Macros*****
                0048
                0049  SALIG   MACRO
                0050          LD      A,E             ; Carga el Ac. con el reg. E
                0051          CALL    SALTTY          ; y envia a la terminal su
                0052          LD      A,'='           ; contenido y un caracter
                0053          CALL    SALTTY          ; de igual "=".
                0054          MEND
                0055
                0056  ESPAC   MACRO
                0057          LD      A,ESP           ; Envio de un caracter
                0058          CALL    SALTTY          ; de espacio a la terminal.
                0059          MEND
*********************************************
**<><><><><><><><><><><><><><><><><><><><><>**
**<>                                       <>**
**<>        PROGRAMA PRINCIPAL             <>**
**<>                                       <>**
**<><><><><><><><><><><><><><><><><><><><><>**
*********************************************
     (FFFF)     0067          IF      ENSAM NE CRO
     (0000)     0068          ORG 000H
     (FFFF)     0069          IF      ENSAM EQ KIT
                0070
Inicializacion del Starter Kit Z80.
Valores de los diversos puertos del Starter Kit:
     (00BC)     0073  DIGLH   EQU     0BCH            ; Puerto del Programador.
```

```
      (0084)      0074  CTC0    EQU     84H             ; Canal 0 del Ctc.
      (0085)      0075  CTC1    EQU     CTC0+1          ; Canal 1 del Ctc.
      (0086)      0076  CTC2    EQU     CTC0+2          ; Canal 2 del Ctc.
      (0087)      0077  CTC3    EQU     CTC0+3          ; Canal 3 del Ctc.
      (0090)      0078  BAUDR   EQU     90H             ; Lectura de Baudaje.
      (0098)      0079  USART   EQU     98H             ; Usart (datos)
                  0080
                  0081  ;Programa Principal para el STARTER KIT.
0000  319A23      0082  RST0:   LD      SP,USTACK1      ; Inicializacion del SP. de
0003  ED736E23    0083          LD      (USTACK),SP     ; Usuario.
0007  318823      0084          LD      SP,STACK        ; Inicializacion de SP.
000A  F3          0085          DI                      ; Deshabilitar Interrupciones.
000B  C35F00    R 0086          JP      INIC1           ; Salta a la inicializacion de
                  0087                                  ; puertos y parametros
                  0088  ;***Rutina para Salvar los diversos Reg. del Mp.
      (0008)      0089          ORG     8
                  0090  RST8:
0008  ED736E23    0091          LD      (USTACK),SP     ; Salvar el SP de usuario
000C  318623      0092          LD      SP,SKREG        ; SP=Pila de registros.
000F                0093          IRP     #VAR,AF,BC,DE,HL,IX,IY
                  0094          PUSH    #               ; Salvado de los registros.
                  0095          MEND
000F  F5          0096+         PUSH    AF              ; Salvado de los registros.
0010  C5          0097+         PUSH    BC              ; Salvado de los registros.
0011  D5          0098+         PUSH    DE              ; Salvado de los registros.
0012  E5          0099+         PUSH    HL              ; Salvado de los registros.
0013  DDE5        0100+         PUSH    IX              ; Salvado de los registros.
0015  FDE5        0101+         PUSH    IY              ; Salvado de los registros.
0017  ED57        0102          LD      A,I             ; Salvado de reg. I
0019  F5          0103          PUSH    AF
001A  08          0104          EX      AF,AF'
001B  D9          0105          EXX
001C                0106          IRP     #VAR,AF,BC,DE,HL
                  0107          PUSH    #VAR            ; Salvado de reg. primos.
                  0108          MEND
001C  F5          0109+         PUSH    AF              ; Salvado de reg. primos.
001D  C5          0110+         PUSH    BC              ; Salvado de reg. primos.
001E  D5          0111+         PUSH    DE              ; Salvado de reg. primos.
001F  E5          0112+         PUSH    HL              ; Salvado de reg. primos.
0020  DD2A6E23    0113          LD      IX,(USTACK)     ; Modificar el valor del
0024  DD23        0114          INC     IX              ; SP de usuario al valor
0026  DD23        0115          INC     IX              ; actual.
0028  DD226E23    0116          LD      (USTACK),IX     ; Modificar el valor del
002C  DD7EFE      0117          LD      A,(IX-2)        ; Contador de Prog. a su
002F  B7          0118          OR      A               ; valor actual.
0030  2003       0119          JR      NZ,RST161
0032  DD35FF      0120          DEC     (IX-1)
0035  DD35FE      0121  RST161  DEC     (IX-2)
0038  3A6D23      0122          LD      A,(BFLAG)       ; Cargar el Ac. con la bandera
003B  B7          0123          OR      A               ; de Puntos de Rup. y si es diferente
003C  281B        0124          JR      Z,BPNO          ; de cero colocar los P. de R. corresp.
003E  47          0125          LD      B,A             ; reg. B=numero de P. de R.
003F  DD215E23    0126          LD      IX,BTAB         ; IX=tabla de P.de R.
0043  DD7E02      0127  Prxbp:  LD      A,(IX+2)        ; Se restaura el codigo de op. original
0046  FECF        0128          CP      0CFH            ; Si el codigo es CFh. no se restaura.
0048  2007        0129          JR      Z,MULTBP
004A  DD6E01      0130          LD      L,(IX+1)        ; Carga HL con la localidad donde
004D  DD6600      0131          LD      H,(IX+0)        ; esta el P. de R.
0050  77          0132          LD      (HL),A          ; Restaura el codigo
0051  DD23        0133  Multbp: INC     IX              ; IX se mueve al siguiente P. de R.
0053  DD23        0134          INC     IX              ; Se repite la eliminacion
0055  DD23        0135          INC     IX              ; de P. de R. hasta llegar al
0057  10EA        0136          DJNZ    PRXBP           ; final de la tabla.
0059  318823      0137  BPNO:   LD      SP,STACK        ; Se reinicializa el SP.
005C  C38D00    R 0138          JP      RECCOM          ; Se salta al Reconocimiento de COM.
                  0139  INIC1
005F  AF          0140          XOR     A               ; A=00
0060  326D23      0141          LD      (BFLAG),A       ; BFLAG=00
0063  3E15        0142          LD      A,CTC16         ; Se programa al CTC0 como
0065  D384        0143          OUT     (CTC0),A        ; temporizador entre 16
0067  DB90        0144          IN      A,(BAUDR)       ; Se lee el  baudaje
0069  0ED8        0145          LD      C,0D8H          ; Cte. para baudaje=300
006B  CB2F        0146  corre:  SRA     A               ; El valor leido se revisa.
```

```
006D  3004      0147          JR      NC,NOCOR        ; No se divide la cte. de tiempo
006F  CB39      0148          SRL     C               ; divide cte. de tiempo /2
0071  18F8      0149          JR      CORRE           ; y verifica si se vuelve a
                0150                                  ; dividir.
0073  D3B4      0151  Nocor:  OUT     (CTC0),A        ;  y se envia
0075  213704    0152          LD      HL,INICTAB      ; Apuntador a tabla de inic.
0078  0E99      0153          LD      C,USART+1       ; del Usart.
007A  0605      0154          LD      B,05            ; Contador para numero de palabras
007C  EDB3      0155          OTIR
                0156          ELSE
                0157
                0158  ;INICIALIZACION DEL MKZ80
                0159  STACK   EQU     17BEH
                0160  CTC0    EQU     20H
                0161  USART   EQU     80H
                0162  INIC    DI
                0163          LD      SP,STACK
                0164          LD      C,CTC0
                0165          LD      HL,INICTAB      ;APUNTADOR A TABLA DE INIC.
                0166          LD      B,03            ;CONTADOR PARA PROG CTC0
                0167          OTIR                    ;PROGR. CTC CANAL 0
                0168          LD      C,CTC0+1
                0169          LD      B,02
                0170          OTIR                    ;PROGR. CTC CANAL 1
                0171          LD      A,05H
                0172          OUT     (CTC0+2),A      ;PROGR. CANAL2 CONTROL
                0173          LD      A,(CTET)
                0174          OUT     (CTC0+2),A      ;CTE DE TEMP. PARA EL USART
                0175          LD      C,CTC0+3
                0176          LD      B,02
                0177          OTIR                    ;PROGR. CTC CANAL 3
                0178          LD      C,USART+1       ;CTRL DEL USART
                0179          LD      B,05
                0180          OTIR                    ;PROGR. DEL USART (8251)
                0181          ENDIF
                0182          ELSE
                0183          REL
                0184  Prin:   LD      SP,STACK        ; Se inicializa el SP.
                0185          ENDIF
                0186  ;*************************************************
                0187  ;RUTINA DE DESPLIGUE DE LETRERO
                0188  ;*************************************************
007E  213C04    0189          LD      HL,Inilet       ; Apuntador a letrero a
                0190                                  ; desplegar.
0081  0642      0191          LD      B,FINAL-Inilet  ; Contador de Numero. de caract.
0083  7E        0192  LETRI   LD      A,(HL)          ; Envio de los caracteres
0084  CD0003H   0193          CALL    SALTTY          ; del letrero de inicio.
0087  23        0194          INC     HL
0088  18F9      0195          DJNZ    LETRI
008A  CD0003H   0196          CALL    CRECO           ; Salto de linea al terminar de
                0197                                  ; desplegar el letrero.
                0198  ;*************************************************
                0199  ;   RUTINA DE RECONOCIMIENTO DE COMANDOS
                0200  ;*************************************************
                0201  RECCOM
008D  CD0000H   0202          CALL    POLENT          ; Entra un caracter
0090  CD0003H   0203          CALL    ECO
0093  D641      0204          SUB     'A'             ; COMPARAR RESPEC. A 'A'
0095  DA2000H   0205          JP      C,ERROR         ; <'A'
0098  FE1A      0206          CP      'Z'- A'+1       ; >'Z'
009A  D22000H   0207          JP      NC,ERROR
009D  87        0208          ADD     A               ; A=2*A
009E  210304    0209          LD      HL,COMTAB       ; APUNT. A TABLA DE COM.
00A1  85        0210          ADD     L               ; SUMAR DESPLAZAMIENTO
00A2  6F        0211          LD      L,A
00A3  3001      0212          JR      NC,REC1         ; SIGUE SI ESTA EN LA
00A5  24        0213          INC     H               ; MISMA PAG.
00A6  5E        0214  REC1    LD      E,(HL)          ; CARGA APUNT.
00A7  23        0215          INC     HL              ; CON DIRECC. DEL COM.
00A8  56        0216          LD      D,(HL)
00A9  EB        0217          EX      DE,HL           ; INTERCAMBIO DE APUNT.
00AA  E9        0218          JP      (HL)            ; SALTA A COM.
                0219
```

```
                        0220  ;************************************
                        0221  ;*          Comando  LISTA          *
                        0222  ;************************************
                        0223  LCOM
00AB  CD0000#           0224          CALL    CO16            ; Entra M1(localidad inicial)
00AE  EB                0225          EX      DE,HL           ; Salva M1
00AF  CD0000#           0226          CALL    CO16            ; Entra M2(localidad final)
00B2  CD0000#           0227          CALL    RESTA           ; Calculo M1-M2
00B5  EB                0228          EX      DE,HL           ; Restauro M1 (HL=M1,DE=M1-M2+1)
00B6  CD0000#           0229  Lcom1:  CALL    SALHL           ; Sale localidad inicial
00B9  E5                0230          PUSH    HL              ; Salva valor apunt.
00BA                    0231          ESPAC                   ; y espacio.
00BA  3E20              0232+         LD      A,ESP           ; Envio de un caracter
00BC  CD0000#           0233+         CALL    SALTTY          ; de espacio a la terminal.
00BF  CD0000#           0234          CALL    CTRLC           ; Se desea abortar el comando?
00C2  7E                0235  Lcom2:  LD      A,(HL)          ; Se envia (HL)
00C3  CD0000#           0236          CALL    BYTTY           ; y un espacio
00C6                    0237          ESPAC
00C6  3E20              0238+         LD      A,ESP           ; Envio de un caracter
00C8  CD0000#           0239+         CALL    SALTTY          ; de espacio a la terminal.
00CB  23                0240          INC     HL              ; Incrementa el apuntador
00CC  1B                0241          DEC     DE              ; y se decrementa el contador.
00CD  7A                0242          LD      A,D             ; Si el contador
00CE  B3                0243          OR      E               ; DE=00 es fin del comnado.
00CF  2805              0244          JR      Z,LFIN
00D1  3E0F              0245          LD      A,0FH           ; Si la proxima localidad que se
00D3  A5                0246          AND     L               ; enviara no termina en 0F se envia el
00D4  20EC              0247          JR      NZ,LCOM2        ; contenido de la siguiente localidad
00D6  E5                0248          PUSH    HL              ; salva valor del apuntador
00D7  CD0000#           0249          CALL    PASCII          ; Despliega contenido ASCII de Memoria.
00DA  CD0000#           0250          CALL    CRECO           ; En caso contrario se salta de linea
00DD  18D7              0251          JR      LCOM1           ; y se envia el valor del apuntador.
00DF  E5                0252  Lfin:   PUSH    HL              ; Salva valor del apuntador
00E0  CD0000#           0253          CALL    PASCII          ; Despliega contenido ASCII de Memoria.
00E3  CD0000#           0254          CALL    CRECO           ; Salto de linea.
00E6  C38D00          R 0255          JP      RECCOM          ; Regresa a Rec. de Comandos.
                        0256  ;************************************
                        0257  ;   COMANDO CORRE                  *
                        0258  ;************************************
                        0259  CCOM
00E9  CD0000#           0260          CALL    CO16            ; Entra nuevo valor para el P.C.
00EC  DD216E23          0261          LD      IX,USTACK       ; Coloca el nuevo P.C.
00F0  DD7500            0262          LD      (IX+0),L        ; en el tope de la pila de
00F3  DD7401            0263          LD      (IX+1),H        ; Usuario.
00F6  EB                0264          EX      DE,HL           ; Salva el nuevo P.C. en DE.
00F7  DD215E23          0265          LD      IX,BTAB         ; IX=tabla de P. de R.
00FB  3A6D23            0266          LD      A,(BFLAG)       ; revisa bandera de P.C.
00FE  B7                0267          OR      A               ; si la bandera es difer. de 00
00FF  2821              0268          JR      Z,POPREG        ; se colocan los los P. de R.
0101  47                0269          LD      B,A             ; B=cont. de P.de R.
0102  DD7E00            0270  Crr:    LD      A,(IX+0)        ; Revisa si el nuevo P.C.
0105  BA                0271          CP      D               ; contiene un P. de R.
0106  2006              0272          JR      NZ,PBP          ; de ser asi no se coloca el
0108  DD7E01            0273          LD      A,(IX+1)        ; P. de R.
010B  EB                0274          CP      E
010C  280C              0275          JR      Z,NOBP
010E  DD6600            0276  Pbp:    LD      H,(IX+0)        ; Se carga la localidad donde
0111  DD6E01            0277          LD      L,(IX+1)        ; se desea colocar el P. de R.
0114  7E                0278          LD      A,(HL)          ; Se salva el codigo de op.
0115  DD7702            0279          LD      (IX+2),A        ; del usuario.
0118  36CF              0280          LD      (HL),@CFH       ; Se coloca el P. de R.
011A  DD23              0281  Nobp:   INC     IX              ; Se revisa la tabla de
011C  DD23              0282          INC     IX              ; P. de R. hasta que
011E  DD23              0283          INC     IX              ; sean colocados todos los
0120  10E8              0284          DJNZ    CRR             ; P. de R.
0122  317023            0285  Popreg: LD      SP,SKREG+16H    ; Se Apunta el SP a la base de
0125  ED736E23          0286          LD      (USTACK),SP     ; la pila de REGISTROS
0129                    0287          IRP     #VAR,HL,DE,BC,AF ; y se restauran todos los valores
                        0288          POP     #VAR            ; de usuario en los reg. del mP.
                        0289          MEND
0129  E1                0290+         POP     HL              ; de usuario en los reg. del mP.
012A  D1                0291+         POP     DE              ; de usuario en los reg. del mP.
012B  C1                0292+         POP     BC              ; de usuario en los reg. del mP.
```

<MONITOR PARA TERMINAL>

```
012C  F1          0293+         POP     AF              ; de usuario en los reg. del mP.
012D  D9          0294          EXX
012E  08          0295          EX      AF,AF'
012F  F1          0296          POP     AF
0130  ED47        0297          LD      I,A
0132                0298          IRP     #VAR,IY,IX,HL,DE,BC,AF
                    0299          POP     #VAR
                    0300          MEND
0132  FDE1        0301+         POP     IY
0134  DDE1        0302+         POP     IX
0136  E1          0303+         POP     HL
0137  D1          0304+         POP     DE
0138  C1          0305+         POP     BC
0139  F1          0306+         POP     AF
013A  ED7B6E23    0307          LD      SP,(USTACK)     ; SP=SP de Usuario.
013E  C9          0308          RET                     ; Se salta a PROGRAMA de Usuario
                    0309
                    0310  ;**************************
                    0311  ;* COMANDO DE DES. DE REG *
                    0312  ;**************************
                    0313  RCOM:
013F  3E0D        0314          LD      A,CR            ; Se salta de linea.
0141  CD0000#     0315          CALL    SALTTY
0144  3E0A        0316          LD      A,LF
0146  CD0000#     0317          CALL    SALTTY
0149  DD21B323    0318          LD      IX,SKREG-3      ; Se apunta a IX a la pila de REG.-3
014D  11B604      0319          LD      DE,TAB3         ; DE=>tabla de caracs. de los REGS.
0150  DD7E01      0320          LD      A,(IX+1)        ; A=registro de banderas(A=F).
0153  CD0000#     0321          CALL    BAND            ; Se despliegan las banderas.
0156  CD0000#     0322          CALL    PINTA           ; Se pinta "A ="
0159  DD7E02      0323          LD      A,(IX+2)        ; A=A(de usuario)
015C  CD0000#     0324          CALL    BYTTY           ; se despliega el contenido de A.
015F  0E02        0325          LD      C,02
0161  0605        0326          LD      B,05            ; contador de reg a desplegar
0163              0327  RC1:     ESPAC
0163  3E20        0328+         LD      A,ESP           ; Envio de un caracter
0165  CD0000#     0329+         CALL    SALTTY          ; de espacio a la terminal.
0168  CD0000#     0330          CALL    PINTA           ; Se pinta el carac. del par de reg.
016B  DD6600      0331          LD      H,(IX+0)        ; se carga en HL el valor de un par
016E  DD6EFF      0332          LD      L,(IX-1)        ; de registros y se envia
0171  CD0000#     0333          CALL    SALHL           ; a la terminal
0174  DD2B        0334          DEC     IX              ; se mueve el apuntador
0176  DD2B        0335          DEC     IX              ; de registros
0178  10E9        0336          DJNZ    RC1             ; hasta que el cont.=0
017A  0D          0337          DEC     C               ; si C=0 se despliega el P.C.
017B  CAAF01    R 0338          JP      Z,RCFIN         ; y es fin del comando.
017E  CD0000#     0339          CALL    PINTA           ; Se pinta "I="
0181  DD7E00      0340          LD      A,(IX+0)        ; A=I
0184  CD0000#     0341          CALL    BYTTY           ; se despliega el contenido de A.
0187  DD2B        0342          DEC     IX              ; se mueve el apuntador
0189  DD2B        0343          DEC     IX              ; de registros
018B  3E0D        0344          LD      A,CR            ; se salta de linea.
018D  CD0000#     0345          CALL    SALTTY
0190  3E0A        0346          LD      A,LF
0192  CD0000#     0347          CALL    SALTTY
0195  0606        0348          LD      B,06
0197  3E20        0349          LD      A,20H           ; se envian 6 espacios.
0199  CD0000#     0350  RC2:     CALL    SALTTY
019C  10FB        0351          DJNZ    RC2
019E  CD0000#     0352          CALL    PINTA           ; se pinta "A'="
01A1  DD7E00      0353          LD      A,(IX+0)        ; A=A'
01A4  CD0000#     0354          CALL    BYTTY           ; se despliega el contenido de A'
01A7  DD2B        0355          DEC     IX              ; se mueve el apuntador
01A9  DD2B        0356          DEC     IX              ; de registros
01AB  0604        0357          LD      B,04            ; Contador de regs. a desplegar.
01AD  1804        0358          JR      RC1             ; se despliegan los pares de reg
                    0359                                  ; restantes.
01AF              0360  Rcfin:   ESPAC
01AF  3E20        0361+         LD      A,ESP           ; Envio de un caracter
01B1  CD0000#     0362+         CALL    SALTTY          ; de espacio a la terminal.
01B4  CD0000#     0363          CALL    PINTA           ; se pinta "PC="
01B7  DD2A5E23    0364          LD      IX,(USTACK)     ; IX=SP de usuario
01BB  DD6601      0365          LD      H,(IX+1)        ; se carga HL con el P.C.
```

```
01BE  DD6E00      0366          LD      L,(IX+0)
01C1  CD0000#     0367          CALL    SALHL           ; Se despliega el P.C
01C4  CD0000#     0368          CALL    CRECO           ; se salta de linea
01C7  C38D00      0369          JP      RECCOM          ; y se regresa al Rec. de COM.
                  0370  ;*********************************
                  0371  ; Comando DE PUNTOS DE RUPTURA.
                  0372  ;*********************************
                  0373  BCOM
                  0374
01CA  DD215E23    0375          LD      IX,BTAB         ; Inic. apuntador a tabla de P. de R.
01CE  3A6D23      0376          LD      A,(BFLAG)       ; Carga bandera de P. de R. y
01D1  B7          0377          OR      A               ; Compara con cero
01D2  280D        0378          JR      Z,BP1           ; y si=00 continua
01D4  FE05        0379          CP      05              ; Si es igual a 5
01D6  281C        0380          JR      Z,BRBP          ; la tabla esta llena
01D8  47          0381          LD      B,A             ; B=contador de P. de R. que existen.
01D9  DD23        0382  Mbp:     INC     IX              ; Se mueve el apuntador hasta
01DB  DD23        0383          INC     IX              ; donde se encuentre una vacante en
01DD  DD23        0384          INC     IX              ; la tabla de P. de R.
01DF  10F8        0385          DJNZ    MBP
01E1  CD0000#     0386  Bp1:    CALL    CO16            ; Entra nueva localidad donde
01E4  DD7400      0387          LD      (IX+0),H        ; se colocara el P. de R. y
01E7  DD7501      0388          LD      (IX+1),L        ; se salva en la tabla .
01EA  3A6D23      0389          LD      A,(BFLAG)       ; Incrementa bandera de
01ED  3C          0390          INC     A               ; P. de R.
01EE  326D23      0391          LD      (BFLAG),A       ; y se salva.
01F1  C38D00      0392          JP      RECCOM          ; Salta al Rec. de Com.
01F4  115E23      0393  Brbp:    LD      DE,BTAB         ; Si la tabla esta
01F7  216123      0394          LD      HL,BTAB+3       ; llena se desecha un
01FA  010F00      0395          LD      BC,015          ; P. de R. y se coloca la
01FD  EDB0        0396          LDIR                    ; bandera de P. de R.
01FF  3E04        0397          LD      A,04            ; en 4. Luego se salta a
0201  326D23      0398          LD      (BFLAG),A       ; Colocar el P. de R.
0204  18C4        0399          JR      BCOM
                  0400
                  0401  ;*********************************
                  0402  ;* COMANDO DE SUSTITUSION DE REG.  *
                  0403  ;*********************************
                  0404  XCOM
0206  DD21B623    0405          LD      IX,SKREG        ; IX es apuntador a Stack de Regs.
020A  21AD04      0406          LD      HL,RTAB         ; HL es apuntador a tabla de sust.
020D  011100      0407          LD      BC,RCON         ; BC es contador de sustitucion.
0210  CD0000#     0408          CALL    POLENT          ; Entra un caracter y se compara con
0213  EDA1        0409  REPITE: CPI                     ; la tabla a la vez que se mueven ambos
0215  DD2B        0410          DEC     IX              ; apuntadores.
0217  E20000#     0411          JP      PO,ERROR        ; Si el caracter no esta en la tabla
021A  20F7        0412          JR      NZ,REPITE       ; se va a error y si esta en la tabla
021C  5F          0413          LD      E,A             ; se salva el caracter en E.
021D  CD0000#     0414          CALL    ECO             ; Se rebota el caracter
0220  CD0000#     0415          CALL    POLENT          ; se lee otro car. y se rebota
0223  CD0000#     0416          CALL    ECO
0226  FE0D        0417          CP      CR              ; Si el car.=CR se sustituye el reg.
0228  282E        0418          JR      Z,SUSR
022A  FE27        0419          CP      27H             ; Si el car.=comillas
022C  2803        0420          JR      Z,SUSPRI        ; se sustituye regs. primos.
022E  C30000#     0421          JP      ERROR           ; Salta a error
0231  CD0000#     0422  SUSPRI  CALL    POLENT          ; Se lee un caracter y
0234  CD0000#     0423          CALL    ECO             ; se rebota.
0237  FE0D        0424          CP      CR              ; si car.<> CR
0239  C20000#     0425          JP      NZ,ERROR        ; se va a Error.
023C              0426          SALIG
023C  7B          0427+         LD      A,E             ; Carga el Ac. con el reg. E
023D  CD0000#     0428+         CALL    SALTTY          ; y envia a la terminal su
0240  3E3D        0429+         LD      A,'='           ; contenido y un caracter
0242  CD0000#     0430+         CALL    SALTTY          ; de igual "=".
0245  DD7EF2      0431          LD      A,(IX-14)       ; Se carga el reg. primo
0248  CD0000#     0432          CALL    BYTTY           ; en A y se despliega su conte.
0248              0433          ESPAC
0248  3E20        0434+         LD      A,ESP           ; Envio de un caracter
024D  CD0000#     0435+         CALL    SALTTY          ; de espacio a la terminal.
0250  CD0000#     0436          CALL    CO16            ; Entra su nuevo valor
0253  DD75F2      0437          LD      (IX-14),L       ; y se sustituye.
0256  185D        0438          JR      FXCOM           ; salta al fin del Com.
```

```
0258                    0439  SUSR:   SALIG
0258  7B                0440+         LD      A,E             ; Carga el Ac. con el reg. E
0259  CD0000#           0441+         CALL    SALTTY          ; y envia a la terminal su
025C  3E3D              0442+         LD      A,'='           ; contenido y un caracter
025E  CD0000#           0443+         CALL    SALTTY          ; de igual "=".
0261  79                0444          LD      A,C             ; Se carga A con el valor del
0262  FE01              0445          CP      01              ; contador. del apunt. de tabla
0264  2834              0446          JR      Z,SUSPC         ; Si es 01 se sust. el PC.
0266  FE04              0447          CP      04              ; Si es 04 se sust. el reg. I.
0268  2804              0448          JR      Z,SUSI
026A  FE09              0449          CP      09H             ; Si es menor de 08 es un reg
026C  3813              0450          JR      C,SUS2          ; de 16 bits.
026E  DD7E00            0451  SUSI:   LD      A,(IX+0)        ; Se carga el cont. del reg.
0271  CD0000#           0452          CALL    BYTTY           ; en A y se despliega.
0274                    0453          ESPAC
0274  3E20              0454+         LD      A,ESP           ; Envio de un caracter
0276  CD0000#           0455+         CALL    SALTTY          ; de espacio a la terminal.
0279  CD0000#           0456          CALL    CO16            ; Entra el nuevo valor
027C  DD7500            0457          LD      (IX+0),L        ; y se sustituye.
027F  1834              0458          JR      FXCOM           ; Se salta al fin del Com.
0281  DD6600            0459  SUS2    LD      H,(IX+0)        ; Se carga el cont. del reg.
0284  DD6EFF            0460          LD      L,(IX-1)        ; en el par HL.
0287  CD0000#           0461          CALL    SALHL           ; y se despliega el cont.
028A                    0462          ESPAC
028A  3E20              0463+         LD      A,ESP           ; Envio de un caracter
028C  CD0000#           0464+         CALL    SALTTY          ; de espacio a la terminal.
028F  CD0000#           0465          CALL    CO16            ; entra el nuevo valor
0292  DD7400            0466          LD      (IX+0),H        ; y se hace la sustitucion.
0295  DD75FF            0467          LD      (IX-1),L
0298  181B              0468          JR      FXCOM           ; Se salta al fin del Com.
029A  DD2A6E23          0469  SUSPC   LD      IX,(USTACK)     ; Se carga IX con el SP de Usuario
029E  DD6601            0470          LD      H,(IX+1)        ; y se carga el valor del PC en
02A1  DD6E00            0471          LD      L,(IX+0)        ; los regs. HL.
02A4  CD0000#           0472          CALL    SALHL           ; Se despliega el valor del PC.
02A7                    0473          ESPAC
02A7  3E20              0474+         LD      A,ESP           ; Envio de un caracter
02A9  CD0000#           0475+         CALL    SALTTY          ; de espacio a la terminal.
02AC  CD0000#           0476          CALL    CO16            ; Entra el nuevo valor
02AF  DD7401            0477          LD      (IX+1),H        ; y se hace la sustitucion.
02B2  DD7500            0478          LD      (IX+0),L
02B5  C38D00            0479  FXCOM:  JP       RECCOM          ; Fin del comando
                        0480                                  ; Regresa al Rec. del Com.
                        0481
                        0482  ;************************************
                        0483  ;COMANDO  INSERTA
                        0484  ;************************************
                        0485  ICOM
02B8  CD0000#           0486          CALL    CO16            ; Entra la localidad desde la
02BB  3E0D              0487          LD      A,CR            ; cual se hara la Insercion
02BD  CD0000#           0488          CALL    SALTTY          ; Se envia un CR a la terminal
02C0                    0489          ESPAC                   ; y un espacio.
02C0  3E20              0490+         LD      A,ESP           ; Envio de un caracter
02C2  CD0000#           0491+         CALL    SALTTY          ; de espacio a la terminal.
02C5  CD0000#           0492  ICOM1   CALL    POLENT          ; Entra un caracter
02C8  FE0D              0493          CP      CR              ; si es=CR o "espacio"
02CA  28F9              0494          JR      Z,ICOM1         ; se desecha el caracter
02CC  CD0000#           0495          CALL    ECO
02CF  FE20              0496          CP      ESP
02D1  28F2              0497          JR      Z,ICOM1
02D3  CD0000#           0498          CALL    CONV            ; Se convierte el caracter a Hex.
02D6  CB21              0499          SLA     C               ; y se corre 4 veces
02D8  CB21              0500          SLA     C               ; hacia la izquierda
02DA  CB21              0501          SLA     C
02DC  CB21              0502          SLA     C
02DE  51                0503          LD      D,C             ; Se salva el caracter Hex. en D.
02DF  CD0000#           0504  ICOM2   CALL    POLENT          ; Entra otro caracter
02E2  FE0D              0505          CP      CR              ; si es=CR o "espacio"
02E4  28F9              0506          JR      Z,ICOM2         ; se desecha el caracter.
02E6  CD0000#           0507          CALL    ECO
02E9  FE20              0508          CP      ESP
02EB  28F2              0509          JR      Z,ICOM2
02ED  CD0000#           0510          CALL    CONV            ; Se convierte el caracter
02F0  7A                0511          LD      A,D             ; a Hex.
```

```
02F1  81        0512          OR      C               ; Se ensambla un valor en A
02F2  77        0513          LD      (HL),A          ; Se guarda el valor hex en
02F3  23        0514          INC     HL              ; memoria y se incrementa el apuntador
02F4  18CF      0515          JR      ICOM1           ; se repite la operacion hasta que entre
                0516                                  ; un caracter que no se HEX.
                0517
                0518  ;*************************************
                0519  ; COMANDO SUBSTITUYE
                0520  ;*************************************
                0521  SCOM
02F6  CD0000#   0522          CALL    CO16            ; Entra localidad de Sustitucion
02F9  CD0000#   0523  SCOM1   CALL    SALHL           ; se despliega la localidad
02FC            0524          ESPAC
02FC  3E20      0525+         LD      A,ESP           ; Envio de un caracter
02FE  CD0000#   0526+         CALL    SALTTY          ; de espacio a la terminal.
0301  7E        0527          LD      A,(HL)          ; se despliega el contenido de la
0302  CD0000#   0528          CALL    BYTTY           ; localidad.
0305            0529          ESPAC
0305  3E20      0530+         LD      A,ESP           ; Envio de un caracter
0307  CD0000#   0531+         CALL    SALTTY          ; de espacio a la terminal.
030A  CD0000#   0532          CALL    POLENT          ; entra un carcter y si es=CR
030D  FE0D      0533          CP      A,CR            ; o espacio se salta a la siguiente
030F  2823      0534          JR      Z,PROX          ; localidad  a sustituir.
0311  CD0000#   0535          CALL    ECO
0314  FE20      0536          CP      ESP
0316  281C      0537          JR      Z,PROX
0318  FE27      0538          CP      27H             ; Si el caracter es (') comilla se
031A  281E      0539          JR      Z,NOCON         ; salta a sustitucion ASCII.
031C  CD0000#   0540          CALL    CONV            ; Si no se convierte el caracter a Hex.
031F  CB21      0541          SLA     C               ; y se corre cuatro veces
0321  CB21      0542          SLA     C               ; a la izquierda.
0323  CB21      0543          SLA     C
0325  CB21      0544          SLA     C
0327  51        0545          LD      D,C             ; Salva el valor Hex. en D
0328  CD0000#   0546          CALL    POLENT          ; entra otro caracter
032B  CD0000#   0547          CALL    ECO             ; se rebota a la termianal.
032E  CD0000#   0548          CALL    CONV            ; se convierte en Hex.
0331  79        0549          LD      A,C             ; y se ensambla el valor
0332  B2        0550          OR      D               ; Hex. en A.
0333  77        0551          LD      (HL),A          ; Se sustituye el valor en memoria
0334  23        0552  PROX    INC     HL              ; y se incrementa el apuntador.
0335  CD0000#   0553          CALL    CRECO           ; se salta de linea.
0338  188F      0554          JR      SCOM1           ; se repite hasta que llegue un valor
                0555                                  ; no Hex.
Sustitucion de valores ASCII:
033A  28        0557  Nocon:  DEC     HL              ; se decrementa el apuntador
033B  CD0000#   0558  Nocon1: CALL    POLENT          ; entra un caracter
033E  FE0D      0559          CP      CR              ; si no es CR se rebota
0340  C40000#   0560          CALL    NZ,ECO
0343  FE27      0561          CP      27H             ; si es (') comilla setermina la
0345  28ED      0562          JR      Z,PROX          ; sustitucion ASCII
0347  23        0563          INC     HL              ; se incrementa el apuntador
0348  77        0564          LD      (HL),A          ; se salva el caracter ASCII
0349  18F0      0565          JR      NOCON1          ; se repite hasta que entre (').
                0566
                0567  ;*************************************
                0568  ;   Comando de Memoria.
                0569  ;*************************************
                0570  MCOM:
034B  CD0000#   0571          CALL    CO16            ; Entra M1 y
034E  EB        0572          EX      DE,HL           ; se salva en los regs. DE
034F  CD0000#   0573          CALL    CO16            ; Entra M2
0352  CD0000#   0574          CALL    RESTA           ; HL=HL-DE+1  (M2-M1+1).
0355  44        0575          LD      B,H             ; BC=HL
0356  4D        0576          LD      C,L
0357  CD0000#   0577          CALL    CO16            ; Entra M3
035A  EB        0578          EX      DE,HL           ; HL=M1, DE=M3 BC=M2-M1+1.
035B  ED80      0579          LDIR                    ; Transfiere M1-M2 a partir M3
035D  C30D00    0580          JP      RECCOM          ; Regresa al Rec. de Com.
                0581  ;*************************************
                0582  ;   Comando de Verificacion de Memoria
                0583  ;*************************************
                0584  VCOM:
```

```
0360  CD0000#     0585          CALL    CO16            ; Entra M1 u
0363  EB          0586          EX      DE,HL           ; se salva en los regs. DE
0364  CD0000#     0587          CALL    CO16            ; Entra M2
0367  CD0000#     0588          CALL    RESTA           ; HL=HL-DE+1  (M2-M1+1).
036A  44          0589          LD      B,H             ; BC=HL
036B  4D          0590          LD      C,L
036C  CD0000#     0591          CALL    CO16            ; Entra M3
036F  EB          0592          EX      DE,HL           ; HL=M1, DE=M3 BC=M2-M1+1.
0370  CD0000#     0593  Vcom1:  CALL    CTRLC           ; Si se oprime ESCAPE se aborta.
0373  1A          0594          LD      A,(DE)          ; Carga a con un dato de M3,(M2-M1+1)
0374  EDA1        0595          CPI                     ; compara A con dato de M1,M2 HL=HL+1
0376  13          0596          INC     DE              ; BC=BC-1 Y DE=DE+1.
0377  E2A603      0597          JP      NV,VFIN         ; Si BC=00 fin de comp.
037A  2002        0598          JR      NZ,DIFER        ; Si Z='off' los datos son diferentes.
037C  18F2        0599          JR      VCOM1           ; Compara el siguiente par de datos de Mem.
037E  1B          0600  Difer:  DEC     DE              ; Si son diferentes se
037F  2B          0601          DEC     HL              ; restauran los apuntadores.
0380  CD0000#     0602          CALL    SALHL           ; Sale el valor del apuntador de M1
0383              0603          ESPAC
0383  3E20        0604+         LD      A,ESP           ; Envio de un caracter
0385  CD0000#     0605+         CALL    SALTTY          ; de espacio a la terminal.
0388  7E          0606          LD      A,(HL)          ; Sale el contenido del apunt. de M1
0389  CD0000#     0607          CALL    BYTTY
038C              0608          ESPAC
038C  3E20        0609+         LD      A,ESP           ; Envio de un caracter
038E  CD0000#     0610+         CALL    SALTTY          ; de espacio a la terminal.
0391  1A          0611          LD      A,(DE)          ; Sale el contenido del apunt. de M3
0392  CD0000#     0612          CALL    BYTTY
0395              0613          ESPAC
0395  3E20        0614+         LD      A,ESP           ; Envio de un caracter
0397  CD0000#     0615+         CALL    SALTTY          ; de espacio a la terminal.
039A  EB          0616          EX      DE,HL           ; Sale el valor del apuntador de M3
039B  CD0000#     0617          CALL    SALHL
039E  EB          0618          EX      DE,HL
039F  CD0000#     0619          CALL    CRECO           ; Salto de linea.
03A2  13          0620          INC     DE              ; Inc. apuntadores.
03A3  23          0621          INC     HL
03A4  18CA        0622          JR      VCOM1           ; Compara el siguiente par de datos de Mem.
03A6  CD0000#     0623  Vfin:   CALL    CRECO           ; Salto de linea.
03A9  3EFC        0624          LD      A,0FCH          ; Envia FC.(Mensaje fin de comparacion.)
03AB  CD0000#     0625          CALL    BYTTY
03AE  3E2E        0626          LD      A,'.'
03B0  CD0000#     0627          CALL    SALTTY
03B3  C38D00      0628          JP      RECCOM          ; Salta al Rec. de Comandos.
      (FFFF)      0629          IF      ENSAM   EQ   KIT
                  0630  ;*************************************
                  0631  ;  Comando de Salida a Puertos
                  0632  ;*************************************
                  0633  Pscom:
03B6  CD0000#     0634          CALL    CO16            ; Entra valor del puerto
03B9  4D          0635          LD      C,L             ; en Hexadecimal.
03BA  7D          0636          LD      A,L
03BB  FE84        0637          CP      84H             ; Si el puerto es el CTC0
03BD  CA0000#     0638          JP      Z,ERROR         ; se va a error.
03C0  CD0000#     0639          CALL    CO16            ; entra valor para enviar
03C3  ED69        0640          OUT     (C),L           ; a el puerto y se envia.
03C5  C38D00      0641          JP      RECCOM          ; regresa al rec. de Com.
                  0642  ;*************************************
                  0643  ;   Comando de entrada de Puerto
                  0644  ;*************************************
                  0645  Pecom:
03C8  CD0000#     0646          CALL    CO16            ; Entra el valor del puerto
03CB  4D          0647          LD      C,L
03CC  ED78        0648          IN      A,(C)           ; Lee el valor del puerto
03CE  CD0000#     0649          CALL    BYTTY           ; y se envia a la terminal
03D1  CD0000#     0650          CALL    CRECO           ; salto de linea.
03D4  C38D00      0651          JP      RECCOM          ; Regresa al rec. de Com.
                  0652  ;*************************************
                  0653  ; Comando DE PROGRAMACION
                  0654  ;*************************************
                  0655  PCOM
03D7  CD0000#     0656          CALL    CO16            ; Entra M1 u
03DA  EB          0657          EX      DE,HL           ; se salva en los regs. DE
```

```
03DB  CD0000#            CALL   C016           ; Entra M2
03DE  CD0000#            CALL   RESTA          ; HL=HL-DE+1  (M2-M1+1).
03E1  44                 LD     B,H            ; BC=HL
03E2  4D                 LD     C,L
03E3  CD0000#            CALL   C016           ; Entra M3
03E6  EB                 EX     DE,HL          ; HL=M1, DE=M3 BC=M2-M1+1.
03E7  3E25     Pcom1:    LD     A,25H          ; Se programa al CTC2 como
03E9  D386               OUT    (CTC2),A       ; temporizador.
03EB  3ECB               LD     A,203D         ; CT de tiempo=203 decimal.
03ED  D386               OUT    (CTC2),A
03EF  3E80               LD     A,80H          ; Se habilita el
03F1  D38C               OUT    (DIGLH),A      ; Prog. de Eproms.
03F3  EDA0               LDI                   ; Se envia el dato a la mem.
03F5  3E00               LD     A,00           ; se deshabilita el prog.
03F7  D38C               OUT    (DIGLH),A      ; de Eproms.
03F9  3E03               LD     A,03           ; Se para el canal CTC2.
03FB  D386               OUT    (CTC2),A
03FD  EAE703             JP     PE,PCOM1       ; Si contador<>00 regresa a prog.
0400  C38D00             JP     RECCOM         ; Salta a rec. de Com.
                         ELSE
                         JP     0000
                         ENDIF
               ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
               ;                                 TABLAS
               ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                         LIST   NOGEN
      (0000)             IF     ENSAM   EQ  CRO
                         DATA
                         ENDIF

               COMTAB
0403  0000#              DW     ERROR          ; A ERROR
0405  CA01               DW     BCOM           ; B COM B.P.
0407  E900               DW     CCOM           ; C COM CORRE
0409  0000#              DW     ERROR          ; D ERROR
      (0000)             IF     ENSAM NE  KIT
                         DW     ERROR          ; E ERROR
                         ELSE
040B  C803               DW     PECOM          ; E COM ENTRADA
                         ENDIF
040D  0000#              DW     ERROR          ; F ERROR
040F  0000#              DW     ERROR          ; G ERROR
0411  0000#              DW     ERROR          ; H ERROR
0413  8802               DW     ICOM           ; I COM INSERTA
0415  0000#              DW     ERROR          ; J ERROR
0417  0000#              DW     ERROR          ; K ERROR
0419  AB00               DW     LCOM           ; L COM LISTA
041B  4B03               DW     MCOM           ; M COM MOVER MEM.
041D  0000#              DW     ERROR          ; N ERROR
      (0000)             IF     ENSAM    NE KIT
                         DW     ERROR          ; O ERROR
                         DW     ERROR          ; P ERROR
                         ELSE
041F  8603               DW     PSCOM          ; O COM SALIDA.
0421  D703               DW     PCOM           ; P COM PROGR.
                         ENDIF
0423  0000#              DW     ERROR          ; Q ERROR
0425  3F01               DW     RCOM           ; R COM REG.
0427  F602               DW     SCOM           ; S COM SUBS.
0429  0000#              DW     ERROR          ; T ERROR
042B  0000#              DW     ERROR          ; U ERROR
042D  6803               DW     VCOM           ; V COM VERF.
042F  0000#              DW     ERROR          ; W ERROR
0431  8602               DW     XCOM           ; X COM CREG.
0433  0000#              DW     ERROR          ; Y ERROR
0435  0000#              DW     ERROR          ; Z ERROR

      (FFFF)             IF     ENSAM  NE  CRO
      (0000)             IF     ENSAM  EQ  MK
               INICTAB
                         DB     90H,CTCPOS,01H  ;VEC. INT,CONTROL Y CTE. DE TIEMPO.
                         DB     CTCNEC,01H              ;CONTROL Y CTE. DE TIEMPO
```

```
                        0731            DB      CTCP0S,01H              ;CONTROL Y CTE. DE TIEMPO
                        0732            DB      80H,00H,40H,4EH         ;CONTROL DEL USART
                        0733            DB      25H                     ;CONTROL DEL USART
                        0734            ELSE
                        0735    INICTAS
0437  80004004          0736            DB      80H,00H,40H,4D
0438  25                0737            DB      25H
                        0738            ENDIF
                        0739            ENDIF
                        0740    ; Letrero de inicio:
                        0741    Inilet:
043C  1B2A204D          0742            DB      1BH,'*',ESP,'MONITOR PARA TERMINAL VER 1.0',CR,LF
      4F4E4954
      4F522050
      41524120
      5445524D
      494E414C
      20564552
      20312E30
      0D0A
045E  2044494D          0743            DB      ESP,'DIME 1983',CR,LF
      45203139
      38330D0A
046A  20476572          0744            DB      ESP,'Gerardo Nuñez Pittu'
      61726446F
      204E7526
      657A2050
      69747479
      (047E)            0745    final   equ     $
                        0746    ;Tabla de caracteres de las banderas del Z80:
047E  535A0048          0747    FTAB    DB      'SZ',NULL,'H'
0482  00564E43          0748            DB      NULL,'VNC'
                        0749    ; tabla de caracteres de los diversos regs. del Z80:
0486  20412042          0750    TAB3    DB      ' A BC DE HL IX',0,'IY',0,'  I'
      43204445
      20484C20
      49580049
      59002020
      49
0498  204127            0751            DB      ' A',27H
049E                    0752            IRP     #REG,BC,DE,HL
                        0753            DB      '#REG',27H
                        0754            MEND
04A7  53500050          0758            DB      'SP',00,'PC',00
      4300
                        0759    ; Tabla para sustitucion de registros:
04AD  41464243          0760    RTAB    DB      'AFBCDEHLX',0,'Y',0,'I',0,'SP'
      4445484C
      58005900
      49005350
                        0761    ; Contador para la tabla anterior:
      (0011)            0762    RCON    EQU     $-RTAB+1
                        0763            ABS
      (235C)            0764            ORG     23C0H-100D
                        0765    ; Valores para Mem. RAM.
235C  (0001)            0766    BASE    DS      1
235D  (0001)            0767    LONG    DS      1
                        0768    ; Tabla de Puntos de Ruptura:
235E  (000F)            0769    BTAB    DS      15
                        0770    ; Bandera de Puntos de Ruptura:
236D  (0001)            0771    BFLAG   DS      1
                        0772    ; Localidad para guardar el Stack de Usuario:
236E  (0002)            0773    USTACK  DS      2
                        0774    ; Stack de registros:
2370  (0016)            0775            DS      16H
      (2386)            0776    SKREG   EQU     $
                        0777    ; Stack de Usuario:
2386  (0014)            0778            DS      20
      (239A)            0779    USTACK1 EQU     $
                        0780    ; Stack del Monitor:
239A  (001E)            0781            DS      30
      (2388)            0782    STACK   EQU     $
                        0783    ;NOTA:el area reservada con DS estara en RAM u el resto en ROM.
```

```
2388  (0000)      0784          END

Errors             0
Range Count        4

Symbol      Value  Defn  References

BAND      X 0000#  0020  0321
BASE        235C   0766
BAUDR       0090   0078  0144
BCOM        01CA   0373  0399 0691
BFLAG       236D   0771  0122 0141 0266 0376 0389 0391 0398
BP1         01E1   0386  0378
BPNO        0059   0137  0124
BRBP        01F4   0393  0380
BTAB        235E   0769  0126 0265 0375 0393 0394
BYTTY     X 0000#  0020  0236 0324 0341 0354 0432 0452 0528 0607 0612 0625 0649
CCOM        00E9   0259  0692
CO16      X 0000#  0020  0224 0226 0260 0386 0436 0456 0465 0476 0486 0522 0571
                         0573 0577 0585 0587 0591 0634 0639 0646 0656
                         0658 0662
COMTAB      0483   0689  0209
CONV      X 0000#  0020  0498 0510 0540 0548
corre       0068   0146  0149
CR          008D   0035  0314 0344 0417 0424 0487 0493 0505 0533 0559 0742 0743
CRECO     X 0000#  0021  0196 0250 0254 0368 0553 0619 0623 0650
CRO         0001   0030  0067 0685 0726
Crr         0102   0270  0284
CTC0        0084   0074  0075 0076 0077 0143 0151
CTC1        0085   0075
CTC16       0015   0040  0142
CTC2        0086   0076  0665 0667 0674
CTC256      0035   0041
CTC3        0087   0077
CTCNEG      00C5   0039
CTCPOS      00D5   0038
CTRLC     X 0000#  0021  0234 0593
CTRLG       0007   0036
DIFER       037E   0600  0598
DIGLH       008C   0073  0669 0672
ECO       X 0000#  0021  0203 0414 0416 0423 0495 0507 0535 0547 0560
ENSAM       0000   0044  0067 0069 0629 0685 0694 0700 0726 0727
ERROR     X 0000#  0023  0205 0207 0411 0421 0425 0638 0690 0693 0699 0700
                          0701 0703 0704 0707 0715 0718 0719 0721 0723 0724
ESP       E 0020   0024 #0033 0232 0238 0328 0361 0434 0454 0463 0474 0498
                         0496 0508 0525 0530 0536 0604 0689 0614 0742 0743
                         0744
ESPAC       Macro  0056  0231 0237 0327 0360 0433 0453 0462 0473 0489 0524
                         0529 0603 0608 0613
FINAL       047E   0745  0191
FTAB      E 047E   0023 #0747
FXCOM       0285   0479  0438 0458 0468
ICOM        0288   0485  0702
ICOM1       02C5   0492  0494 0497 0515
ICOM2       02DF   0504  0506 0509
IGUAL       003D   0032
INIC1       005F   0139  0086
INICTAB     0437   0735  0152
Inilet      043C   0741  0189 0191
KIT         0008   0029  0044 0069 0629 0694 0700
LCOM        00A8   0223  0705
Lcom1       0086   0229  0251
Lcom2       00C2   0235  0247
LETR1       0083   0192  0195
LF          000A   0037  0316 0346 0742 0743
LFIN        00DF   0252  0244
LONG      E 2350   0024 #0767
Mbp         01D9   0382  0385
MCOM        0348   0578  0706
MK          0002   0031  0727
MULTBP      0051   0133  0129
NOBP        011A   0231  0275
```

<MONITOR PARA TERMINAL>

```
Symbol      Value  Defn  References

NOCON       033A   0557  0539
Nocon1      033B   0558  0565
NOCOR       0073   0151  0147
NULL        0000   0028  0747 0748
PASCII   X  0000#  0025  0249 0253
P8P         010E   0276  0272
PCOM        03D7   0655  0713
Pcom1       03E7   0664  0675
Pecom       03C8   0645  0697
PINTA    X  0000#  0021  0322 0330 0339 0352 0363
POLENT   X  0000#  0022  0202 0408 0415 0422 0492 0504 0532 0546 0558
POPREG      0122   0285  0268
PROMPT   E  003E   0024 #0034
PROX        0334   0552  0534 0537 0562
Prxbp       0043   0127  0136
Pscom       03B6   0633  0712
RC1         0163   0327  0336 0358
RC2         0199   0350  0351
RCFIN       01AF   0360  0338
RCOM        013F   0313  0716
RCON        0011   0762  0407
REC1        00A6   0214  0212
RECCOM   E  008D   0023 #0201 0138 0255 0369 0392 0479 0500 0628 0641 0651 0676
REPITE      0213   0409  0412
RESTA    X  0000#  0022  0227 0574 0588 0659
RST0        0000   0082
RST161      0035   0121  0119
RST8        0008   0090
RTAB        04AD   0768  0406 0762
SALHL    X  0000#  0022  0229 0333 0367 0461 0472 0523 0602 0617
SALI6       Macro  0049  0426 0439
SALTTY   X  0000#  0022  0193 0233 0239 0315 0317 0329 0345 0347 0358 0362
                         0428 0430 0435 0441 0443 0455 0464 0475 0488 0491
                         0526 0531 0605 0610 0615 0627
SCOM        02F6   0521  0717
SCOM1       02F9   0523  0554
SKREG       2386   0776  0092 0285 0318 0401
STACK    E  2388   0023 #0782 0084 0137
SUS2        0281   0459  0450
SUSI        026E   0451  0448
SUSPC       029A   0469  0446
SUSPRI      0231   0422  0420
SUSR        0238   0439  0418
TA83        0486   0750  031F
USART    E  0098   0025 #0079 0153
USTACK      236E   0773  0083 0091 0113 0115 0261 0286 0307 0364 0469
USTACK1     239A   0779  0082
VCOM        0360   0584  0720
Vcom1       0370   0593  0599 0622
VFIN        03A6   0623  0597
XCOM        0206   0484  0722
```

APENDICE IV.

Listado del Programa de Transferencia.

```
                        ;**********************************
                        ;*                                *
                        ;*    Programa para Transferencia *
                        ;*      de Archivos tipo HEX al   *
                        ;*      Equipo Starter Kit Z80.   *
                        ;*                                *
                        ;**********************************
                        0009
                        0010  ; Rutinas Externas
                        0011   EXT    AHEX            ; Rutina de conversion ASCII a Hex.
                        0012           EXT     FNAME           ; Rutina para establecer un FCB.
                        0013           EXT     ZOPN            ; Rutina para abrir un archivo existente.
                        0014           EXT     BCDIN           ; Rutina de entrada de valor BCD.
                        0015           EXT     ZIOER           ; Rutina para mensaje de error.
                        0016           EXT     ABORT           ; Rutina para abortar un programa.
                        0017           EXT     GCHAR           ; Rutina para extraer un caracter de
                        0018                                   ; disco u otro dispositivo.
                        0019  ; Definicion de valores:
      (000D)            0020  CR       EQU     0DH             ; ASCII Retorno de carro o cursor.
      (000A)            0021  LF       EQU     0AH             ; ASCII Salto de Linea.
      (0007)            0022  CMP      EQU     07H             ; ASCII Campana.
      (001B)            0023  ESC      EQU     1BH             ; ASCII Escape.
      (0008)            0024  CTRLH    EQU     08H             ; ASCII Control-H (retroceso)
      (001A)            0025  CTRLZ    EQU     1AH             ; ASCII Control-Z (fin de archivo en CDOS).
      (0020)            0026  SPC      EQU     20H             ; ASCII Espacio.
      (005C)            0027  defFCB1  EQU     5CH             ; CDOS FCB1 por default.
      (006C)            0028  defFCB2  EQU     6CH             ; CDOS FCB2 por default.
                        0029
      (1000)            0030          ORG     1000H
                        0031  ; Inicializacion de Parametros
                        0032  Begin:
1000  311615            0033          LD      SP,STACK        ; Inicializacion del SP.
1003  3A5D00            0034          LD      A,(DefFCB1+1)   ; Ac=primer caracter del nombre del archivo
1006  211212            0035          LD      HL,ERR1MSJ      ; "Nombre de archivo ilegal"
1009  FE20              0036          CP      SPC             ; Verifica nombre de archivo en blanco
100B  CAF310            0037          JP      Z,ERROUT        ; y si es imprime mensaje de error
100E  113111            0038          LD      DE,LETR         ; Apuntador de mensaje inicial.
1011  0E09              0039          LD      C,09            ; Pinta letrero
1013  CD0500            0040          CALL    05H
1016  CD0000#           0041          CALL    BCDIN           ; Entra numero de pagina inicial
1019  7D                0042          LD      A,L             ; y se guarda en la localidad
101A  320712            0043          LD      (dato),A        ; "dato".
101D  11A611            0044          LD      DE,LETR1        ; Pinta segundo letrero.
1020  0E09              0045          LD      C,09
1022  CD0500            0046          CALL    05H
1025  11FB11            0047          LD      DE,ENTBUF       ; Apuntador a buffer de entrada.
1028  0E0A              0048          LD      C,0AH           ; Entrada de parametros
102A  CD0500            0049          CALL    05H
102D  3E0D              0050          LD      A,CR            ; Envia un CR y dos LF.
102F  CD0511            0051          CALL    SALCAR
1032  3E0A              0052          LD      A,LF
1034  CD0511            0053          CALL    SALCAR
1037  CD0511            0054          CALL    SALCAR
103A  3E1B              0055          LD      A,ESC           ; Se envia secuencia de
103C  CD0511            0056          CALL    SALCAR          ; caracteres de control
103F  3E2A              0057          LD      A,'*'           ; Esc,'*' y 'I'.
1041  CD0511            0058          CALL    SALCAR
1044  3E49              0059          LD      A,'I'
1046  CD0511            0060          CALL    SALCAR
1049  3AFC11            0061          LD      A,(ENTBUF+1)    ; Cargar Ac. con long. actual
104C  FE05              0062          CP      05H             ; del buffer de entrada; verificar
104E  FA6A10            0063          JP      M,MENOR         ; si es >4 o <4.
1051  D604              0064          SUB     04H             ; Si la long. actual >4
1053  21FD11            0065          LD      HL,ENTBUF+2     ; Apuntador a Buffer de entrada
1056  1600              0066          LD      D,00H           ; y se le suma la diferencia
1058  5F                0067          LD      E,A             ; al apuntador.
1059  19                0068          ADD     HL,DE
105A  0604              0069          LD      B,04            ; Se envian los ultimos cuatro
105C  7E                0070  Nn2:    LD      A,(HL)          ; caracteres que entraron.
105D  CD0511            0071          CALL    SALCAR
1060  23                0072          INC     HL
1061  10F9              0073          DJNZ    NN2
1063  3E0D              0074          LD      A,CR
```

```
<*PROGRAMA DE TRANSFERENCIA*>

1065  CD0511      0075            CALL    SALCAR
1068  1810        0076            JR      TRAN
106A  21FD11      0077  Menor:    LD      HL,ENTBUF+2     ; En caso que la long.<=4 se
106D  47          0078            LD      B,A             ; se carga un contador=long.
106E  7E          0079  Nn1:      LD      A,(HL)          ; y se envian los caracteres
106F  CD0511      0080            CALL    SALCAR          ; que entraron.
1072  23          0081            INC     HL
1073  10F9        0082            DJNZ    NN1
1075  3E0D        0083            LD      A,CR            ; Se envia un caracter [CR]
1077  CD0511      0084            CALL    SALCAR          ; para que se ejecute el com. I.
107A  215C00      0085  Tran:     LD      HL,DefFCB1      ; Se construye un nuevo FCB.
107D  116E12      0086            LD      DE,IXFCB
1080  3E48        0087            LD      A,'H'           ; Se utiliza la extension HEX
1082  015845      0088            LD      BC,'EX'         ; para el archivo que se leyo.
1085  CD0000#     0089            CALL    FNAME
1088  CD0000#     0090            CALL    ZOPN            ; Se abre el archivo de entrada.
108B  CC0000#     0091            CALL    Z,ZIOER         ; En caso de error imprime mensaje
                  0092                                    ; de error y regresa a CDOS
                  0093  ;*********************************
                  0094  ;*       Programa Principal      *
                  0095  ;*********************************
                  0096
108E  3A0712      0097            LD      A,(DATO)        ;Carga Ac. con #de pagina
1091  57          0098            LD      D,A             ;Salva #de pagina en D
1092  CB22        0099            SLA     D               ;D=D*2+1
1094  14          0100            INC     D
1095  1E01        0101            LD      E,001H
1097  CDF610      0102  Lee1:     CALL    Carent          ; Verifica si comienza un bloque
109A  FE3A        0103            CP      ':'
109C  20F9        0104            JR      NZ,Lee1
109E  CDF610      0105  Lee2:     CALL    Carent          ; Los dos caracteres despues del
10A1  320F12      0106            LD      (Car1),A        ; comienzo de bloque dan la longitud
10A4  CDF610      0107            CALL    Carent          ; del bloque.
10A7  321012      0108            LD      (Car1+1),A
10AA  010F12      0109            LD      BC,Car1
10AD  CD0000#     0110            CALL    AHEX            ; La longitud se convierte a Hexa.
10B0  7D          0111            LD      A,L
10B1  FE00        0112            CP      00
10B3  2836        0113            JR      Z,EOF
10B5  45          0114            LD      B,L             ; Carga contador de bloque
10B6  CB20        0115            SLA     B               ; Multiplica reg. B por 2 ya que son
10B8  CDF610      0116            CALL    Carent          ; Se desechan 6 bytes despues de la long
10BB  CDF610      0117            CALL    Carent
10BE  CDF610      0118            CALL    Carent
10C1  CDF610      0119            CALL    Carent
10C4  CDF610      0120            CALL    Carent
10C7  CDF610      0121            CALL    Carent
10CA  CDF610      0122  Lee4:     CALL    Carent
10CD  4F          0123            LD      C,A
10CE  3A6D12      0124            LD      A,(BAND)
10D1  FE41        0125            CP      'A'
10D3  2808        0126            JR      Z,LEE3
10D5  1D          0127            DEC     E
10D6  200F        0128            JR      NZ,LEE5
10D8  15          0129            DEC     D
10D9  280C        0130            JR      NZ,LEE5
10DB  3E41        0131            LD      A,'A'
10DD  326D12      0132            LD      (BAND),A
10E0  79          0133  Lee3:     LD      A,C
10E1  CD0511      0134            CALL    Salcar
10E4  CD1011      0135            call    ctrlc
10E7  10E1        0136            Lee5:   DJNZ    Lee4
10E9  18AC        0137            JR      Lee1
                  0138  ;Rutina de fin de archivo
                  0139
10EB  214E12      0140  EOF:     LD      HL,EOFMs;       ; Apuntador a mensaje de Fin de archivo
                  0141                                    ; Se pinta el mensaje y regresa a CDOS
10EE  1803        0142            JR      Errout
10F0  212B12      0143  Abrtc:   LD      HL,ABMs;
                  0144  ;* Rutina para pintar mensaje de error y regresar a CDOS.
10F3  C3000#      0145  Errout:   JP      Abort           ; Aborta programa y regresa a CDOS
                  0146
                  0147  ;*************************************
```

```
                    0148  ;  RUTINA DE ENTRADA DE CARACTERES
                    0149  ;*********************************
                    0150  Caren#:
10F6  C5            0151          PUSH    BC              ; Salvar los registros.
10F7  D5            0152          PUSH    DE
10F8  116E12        0153          LD      DE,IXFCB        ; Se carga el apuntador
10FB  CD0000#       0154          CALL    GCHAR           ; al FCB y se trae un caracter
10FE  FE1A          0155          CP      CTRLZ           ; del archivo y si=fin de archivo
1100  28E9          0156          JR      Z,EOF           ; se va al control para fin
1102  D1            0157          POP     DE              ; de archivo
1103  C1            0158          POP     BC              ; Restaurar los reg.
1104  C9            0159          RET
                    0160  ;*************************************
                    0161  ;Rutina de Salida de un Caracter
                    0162  ;*************************************
1105  D5            0163  Salcar  PUSH    DE              ; Salvar registros
1106  C5            0164          PUSH    BC
1107  5F            0165          LD      E,A             ; Se envia el contenido
1108  0E02          0166          LD      C,02H           ; de Ac. a la terminal.
110A  CD0500        0167          CALL    05H
110D  C1            0168          POP     BC              ; Se restauran los reg.
110E  D1            0169          POP     DE
110F  C9            0170          RET
                    0171  ;*************************************
                    0172  ;       Rutina para abortar el Programa
                    0173  ;*************************************
                    0174  Ctrlc#:
1110  F5            0175          PUSH    AF              ; Salva registros
1111  C5            0176          PUSH    BC
1112  0E0B          0177          LD      C,11            ; ver el status del teclado
1114  CD0500        0178          CALL    5
1117  FE00          0179          CP      00              ; y si no se oprimio ninguna
1119  280F          0180          JR      Z,CTRLCF        ; tecla se regresa.
111B  0E00          0181          LD      C,00H           ; Si se oprimio alguna tecla
111D  CD0500        0182          CALL    5               ; se lee el teclado.
1120  FE1B          0183          CP      1BH             ; el dato de entrada se
1122  CAF010      R 0184          JP      Z,ABRTC         ; compara con 'ESCAPE' o
1125  FE03          0185          CP      03              ; Ctrl-C, con los cuales
1127  CAF010      R 0186          JP      Z,ABRTC         ; se aborta el programa.
112A  C1            0187  Ctrlcf: POP     BC              ; regresa sin afectar cuando no
112B  F1            0188          POP     AF              ; se oprime ninguna tecla o si
112C  C9            0189          RET                     ; la tecla es <> de 'ESCAPE' o Ctrl-C
                    0190
                    0191
```

```
<*PROGRAMA DE TRANSFERENCIA*>
<DATOS>

                        0192  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        0193  ;                       DATOS
                        0194  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                        0195
112D  07070000          0196  BELL:   DEFB    CMP,CMP,00,00
1131  1B2A071B          0197  LETR:   DEFB    ESC,'*',CMP,ESC,'G4Programa de transferencia',LF,CR
1152  1B473454          0198          DEFB    ESC,'G4Teclee el registro (256bytes)'
1172  0A0D1B47          0199          DEFB    LF,CR,ESC,'G4a partir del cual'
1188  20736520          0200          DEFB    ' se hara la transferencia ',ESC,'G0$'
11A6  1B47340D          0201  LETR1:  DEFB    ESC,'G4',CR,LF,'Teclee la Localidad en Hexa. XXXX',LF,CR
11CE  1B473473          0202          DEFB    ESC,'G4se tomaran los ultimos cuatro digitos ',ESC,'G0$'
                        0203  ;Buffer de entrada de origen de Transferencia.
11FB  0A                0204  ENTBUF: DEFB    0AH
11FC  (0008)            0205          DEFS    08H
                        0206  ;buffer de entrada de Contador de registros
1207  (0008)            0207  DATO:   DEFS    08H
                        0208
120F  (0002)            0209  Car1:   DEFS    2
1211  48                0210          DEFB    'H'
1212  4E6F6D62          0211  Err1Ms.:DEFB    'Nombre de archivo ilegal',CR
1228  0A2A2A2A          0212  ABMs.:  DEFB    LF,'*****Transferencia abortada******',CR
124E  0A2A2A2A          0213  EOFMs.: DEFB    LF,'*** Fin de Transferencia  ***',CR
                        0214
                        0215
                        0216  ; Bandera de conteo de bloques.
126D  (0001)            0217  Band:   DEFS    1
                        0218
                        0219  ;  Extension del FCB.
126E  00                0220  IXFCB:  DEFB    0               ; Contador de Byte
126F  (0022)            0221          DEFS    34              ; Banderas y FCB de CDOS
1291  9612              0222          DEFW    IBUFF,0         ; Apuntador a buffer y numero del buffer actual
      0000
1295  04                0223          DEFB    4               ; Numero de buffers
                        0224
                        0225
                        0226  ;  Buffer de entrada.
1296  (0200)            0227  IBUFF:  DEFS    4*80H           ; Buffer de entrada
                        0228
                        0229
                        0230  ;  Espacio para el Stack
1496  (0080)            0231          DEFS    80H             ; Espacio reservado para el
      (1516)            0232  STACK   EQU     $               ; (Stack se va decrementando)
                        0233
1516  (1000)            0234          END     Begin

Errors                   0
Range Count              2
```

<#PROGRAMA DE TRANSFERENCIA#>

Symbol		Value	Defn	References
ABMsi		122B	0212	0143
ABORT	X	0000#	0016	0145
Abrtc		10F0	0143	0184 0186
AHEX	X	0000#	0011	0110
BAND		126D	0217	0124 0132
BCDIN	X	0000#	0014	0041
Begin		1000	0032	0234
BELL		112D	0196	
Car1		120F	0209	0106 0108 0109
Carent		10F6	0158	0102 0105 0107 0116 0117 0118 0119 0120 0121 0122
CMP		0007	0022	0196 0196 0197
CR		000D	0020	0050 0074 0083 0197 0199 0201 0201 0211 0212 0213
ctrlc		1110	0174	0135
CTRLCF		112A	0187	0188
CTRLH		0008	0024	
CTRLZ		001A	0025	0155
dato		1207	0207	0043 0097
defFCB1		005C	0027	0034 0085
defFCB2		006C	0028	
ENTBUF		11FB	0204	0047 0061 0065 0077
EOF		10EB	0140	0113 0156
EOFMsi		124E	0213	0140
ERR1MSJ		1212	0211	0035
ERROUT		10F3	0145	0037 0142
ESC		001B	0023	0055 0197 0197 0198 0199 0200 0201 0202 0202
FNAME	X	0000#	0012	0089
GCHAR	X	0000#	0017	0154
IBUFF		1296	0227	0222
IXFCB		126E	0220	0086 0153
Lee1		1097	0102	0104 0137
Lee2		109E	0105	
LEE3		10E0	0133	0126
Lee4		10CA	0122	0136
LEE5		10E7	0136	0128 0130
LETR		1131	0197	0038
LETR1		11A6	0201	0044
LF		000A	0021	0052 0197 0199 0201 0201 0212 0213
MENOR		106A	0077	0063
Nn1		106E	0079	0082
Nn2		105C	0070	0073
SALCAR		1105	0163	0051 0053 0054 0056 0058 0060 0071 0075 0080 0084 0134
SPC		0020	0026	0036
STACK		1516	0232	0033
TRAN		107A	0085	0076
ZIOER	X	0000#	0015	0091
ZOPN	X	0000#	0013	0090

APENDICE V.

Listado del Programa de Ejemplo.

```
*** RELOJ ***

*****************************
*                           *
*   PROGRAMA DE EJEMPLO     *
*     RELOJ DE TIEMPO       *
*          REAL             *
*                           *
*****************************
                  0008            GLOBAL  ERROR,RECCOM,SALTTY
                  0009            GLOBAL  BYTTY,CTRLC,STACK
                  0010
                  0011  *RELLIB   RUT
                  0012
      (0064)      0013  CTC0     EQU     64H
      (0066)      0014  CTC2     EQU     CTC0+2
      (0025)      0015  CTCONT   EQU     25H
      (2000)      0016            org     2000H
2000  F3          0017            DI                      ; Deshab. int.
2001  3E20        0018            LD      A,HIGH(INTVEC)  ; Cargo el reg. I con la parte
2003  ED47        0019            LD      I,A             ; alta del vector de int.
2005  3EC0        0020            LD      A,LOW(INTVEC)   ; Se envia la parte baja del
2007  D364        0021            OUT     (CTC0),A        ; vector de int. al CTC.
2009  3E25        0022            LD      A,CTCONT        ; Programar CTC2 como contador
2008  D366        0023            OUT     (CTC2),A        ; q que genere interrupciones.
200D  3E60        0024            LD      A,60H
200F  D366        0025            OUT     (CTC2),A        ; Divisor entre 60.
2011  21AA20      0026  START:    LD      HL,TIME         ; Se inicializa el reloj
2014  AF          0027            XOR     A               ; con un valor 0000.
2015  0604        0028            LD      B,4
2017  77          0029  ET1:      LD      (HL),A
2018  23          0030            INC     HL
2019  10FC        0031            DJNZ    ET1
201B  3E41        0032            LD      A,'A'           ; Se inicializa la bandera
201D  32AF20      0033            LD      (TBAND),A       ; de Am-Pm.
2020  ED5E        0034            IM2                     ; Se elige el Modo dos de int. q
2022  FB          0035            EI                      ; Se habilitan interrupciones
2023  76          0036            HALT                    ; El mP. espera a que ocurra
                  0037                                    ; la interrupcion.
                  0038
                  0039
Rutina de Interrupcion:
                  0041
2024  F3          0042  VECC3:    DI                      ; Se deshabilitan Interrupciones.
2025  21AA20      0043            LD      HL,TIME         ; Se carga el apuntador al "reloj"
2028  CD7B28      0044            CALL    DESPLE          ; Rutina para desplegar la hora
2028  7E          0045            LD      A,(HL)          ; se cargan los segundos
202C  C601        0046            ADD     A,01            ; se incrementan
202E  27          0047            DAA                     ; ajuste dec.
202F  77          0048            LD      (HL),A          ; se carga el valor actual
2030  FE60        0049            CP      60H             ; Segundos=60?
2032  2039        0050            JR      NZ,SIGUE        ; Si es menor regresa.
2034  AF          0051            XOR     A               ; Si seg=60 seg=00
2035  77          0052            LD      (HL),A          ; se carga el valor actual
2036  23          0053            INC     HL              ; Apuntador a min.
2037  7E          0054            LD      A,(HL)          ; se cargan los minutos
2038  C601        0055            ADD     A,01            ; se inc       rementan
203A  27          0056            DAA                     ; ajuste dec.
203B  77          0057            LD      (HL),A          ; carga el valor actual
203C  FE60        0058            CP      60H             ; Min=60?
203E  202D        0059            JR      NZ,SIGUE        ; si es menor regresa.
2040  AF          0060            XOR     A
2041  77          0061            LD      (HL),A
2042  23          0062            INC     HL
2043  7E          0063            LD      A,(HL)
2044  C601        0064            ADD     A,01
2046  27          0065            DAA
2047  77          0066            LD      (HL),A
2048  FE13        0067            CP      13H             ; Horas=12?
204A  2021        0068            JR      NZ,SIGUE
204C  AF          0069            XOR     A
204D  C601        0070            ADD     A,01
204F  77          0071            LD      (HL),A
2050  3AAF20      0072            LD      A,(TBAND)
2053  FE41        0073            CP      'A'
```

```
2055  2811        0074          JR      Z,PONP          ; Tband=AM?
2057  3E41        0075          LD      A,'A'
2059  32AF20      0076          LD      (TBAND),A
205C  23          0077          INC     HL
205D  7E          0078          LD      A,(HL)
205E  C601        0079          ADD     A,01
2060  27          0080          DAA
2061  77          0081          LD      (HL),A
2062  FE30        0082          CP      30H
2064  28A8        0083          JR      Z,START
2066  1005        0084          JR      SIGUE
2068  3E50        0085  PONP    LD      A,'P'
206A  32AF20      0086          LD      (TBAND),A
206D  FB          0087  SIGUE:  EI                      ; Habilitan interrupciones.
206E  ED4D        0088          RETI                    ; Se regresa al programa principal.
                  0089
                  0090  ;***************************************
                  0091  ;*      Rutina de Desplegado del tiempo.*
                  0092  ;***************************************
                  0093
2070  E5          0094  Desple: PUSH    HL
2071  F5          0095          PUSH    AF
2072  21AD20      0096          LD      HL,TIME+3
2075  3E1B        0097          LD      A,1BH
2077  CD0000#     0098          CALL    SALTTY
207A  3E3D        0099          LD      A,'='
207C  CD0000#     0100          CALL    SALTTY
207F  3E20        0101          LD      A,' '
2081  CD0000#     0102          CALL    SALTTY
2084  3E5E        0103          LD      A,'↑'
2086  CD0000#     0104          CALL    SALTTY
2089  0604        0105          LD      B,04
208B  7E          0106  DESZ1:  LD      A,(HL)
208C  CD0000#     0107          CALL    BYTTY
208F  2B          0108          DEC     HL
2090  3E3A        0109          LD      A,':'
2092  CD0000#     0110          CALL    SALTTY
2095  10F4        0111          DJNZ    DESZ1
2097  3E20        0112          LD      A,' '
2099  CD0000#     0113          CALL    SALTTY
209C  3AAF20      0114          LD      A,(TBAND)
209F  CD0000#     0115          CALL    SALTTY
20A2  3E4D        0116          LD      A,'M'
20A4  CD0000#     0117          CALL    SALTTY
20A7  F1          0118          POP     AF
20A8  E1          0119          POP     HL
20A9  C9          0120          RET
                  0121
                  0122  ;%%%%%%%%%%  DATOS  %%%%%%%%%%%%%
20AA  (0005)      0123  TIME:   DS      5
20AF  (0001)      0124  TBAND:  DS      1
20B0  (0010)      0125          DS (10H+((LOW $) AND 0F0H) )-(LOW $)
20C0  C020        0126  INTVEC  DW      VECC0
20C2  C920        0127          DW      VECC1
20C4  CA20        0128          DW      VECC2
20C6  2420        0129          DW      VECC3
                  0130  ;RUTINAS DE INTERRUPCION
20C8  00          0131  VECC0   NOP
20C9  00          0132  VECC1   NOP
20CA  00          0133  VECC2   NOP
                  0134


Errors            0
Range Count       0
```

*** RELOJ ***

Symbol		Value	Defn	References
BYTTY	X	0000#	0009	0107
CTC0		0064	0013	0014 0021
CTC2		0066	0014	0023 0025
CTCONT		0025	0015	0022
CTRLC	X	0000#	0009	
DESPLE		2078	0094	0044
DESZ1		2098	0106	0111
ERROR	X	0000#	0008	
ET1		2017	0029	0031
INTVEC		20C0	0126	0018 0020
PONP		2068	0085	0074
RECCOM	X	0000#	0008	
SALTTY	X	0000#	0008	0090 0100 0102 0104 0110 0113 0115 0117
SIGUE		206D	0087	0050 0059 0068 0084
STACK	X	0000#	0009	
START		2011	0026	0083
TBAND		20AF	0124	0033 0072 0076 0086 0114
TIME		20AA	0123	0026 0043 0096
VECC0		20C8	0131	0126
VECC1		20C9	0132	0127
VECC2		20CA	0133	0128
VECC3		2024	0042	0129

```
B.SCREEN RELOJ.Z80

>Exit: Update

3K written to RELOJ.Z80

B.ASMB RELOJ.@@@
CROMEMCO Z80 Macro Assembler version 03.07

Errors              0
Range Count         0

End of Assembly

B.LINK RELOJ,RELOJ/N/X/E

Link version 03.37

Data    2000    2237

C0000   2037    341


B.TYPE RELOJ.HEX
:1C200000F33E20ED473EC0D3643E25D3663E60D36621AA20AF0604772310FC3E0F
:1C201C004132AF20ED5EFB76F321AA20CD70207EC6012777FE602039AF77237E0E
:1C203800C6012777FE60202DAF77237EC6012777FE132021AFC601773AAF20FE0A
:1C20540041281 13E4132AF20237EC6012777FE3028AB18053E5032AF20FBED4D8E
:1C207000E5F521AD203E1BCD82213E3DCD82213E20CD82213E5ECD822106047E76
:1C208C00CDEF202B3E3ACD822110F43E20CD82213AAF20CD82213E4DCD8221F112
:1C20A800E1C900000000000000000000000000000000000000000C820C920A1
:0720C400CA2024200000000E2
:0000000000

B.TR4 RELOJ
*G4Programa de transferencia
G4Tecleo el registro (256bytes)
G4a partir del cual se hara la transferencia G00G4
recleo la localidad en Hexa. XXXX
G4se tomaran los ultimos cuatro digitos G02000

*3200CFD423EC0D3643E25D3663E6
                                    *****Transferencia abortada******

*.
```

```
* MONITOR PARA TERMINAL VER 1.0
 DIME 1983
 Gerardo Nuñez Pibbs
>L2000,20CA
>2000 F3 3E 20 ED 47 3E C0 D3 64 3E 25 D3 66 3E 60 D3    s> mG>@Sd>%S`S
>2010 66 21 AA 20 AF 06 04 77 23 10 FC 3E 41 32 AF 20    f!* /..w#.|>A2/
>2020 ED 5E FB 76 F3 21 AA 20 CD 70 20 7E C6 01 27 77    m^{vs!* Mp ~F.'w
>2030 FE 60 20 39 AF 77 23 7E C6 01 27 77 FE 60 20 2D    ~` 9/w#~F.'w~`
>2040 AF 77 23 7E C6 01 27 77 FE 13 20 21 AF C6 01 77    /w#~F.'w~. !/F.w
>2050 3A AF 20 FE 41 28 11 3E 41 32 AF 20 23 7E C6 01    :/ ~A(.>A2/ #~F.
>2060 27 77 FE 30 28 AB 18 05 3E 50 32 AF 20 FB ED 4D    'w~0(+..>P2/ {mM
>2070 E5 F5 21 AD 20 3E 1B CD 96 21 3E 3D CD 96 21 3E    eu!- >.M.!>=M.!>
>2080 20 CD 96 21 3E 5E CD 96 21 06 04 7E CD 03 21 2B     M.!>^M.!..~M.!+
>2090 3E 3A CD 96 21 10 F4 3E 20 CD 96 21 3A AF 20 CD    >:M.!.t> M.!:/ M
>20A0 96 21 3E 4D CD 96 21 F1 E1 C9 00 00 00 00 00 00    .!>MM.!qaI......
>20B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
>20C0 C8 20 C9 20 CA 20 24 20 00 00 00    H I J $ ...
>I20D0
>0000000000000000000000000000000000000000000000000000000000000000?*
>L20D0,20EF
>20D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    ................
>20E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 20    ................
>I20EF
>0000.*
>M2000,2020,20D0
>L20D0,20F0
>20D0 F3 3E 20 ED 47 3E C0 D3 64 3E 25 D3 66 3E 60 D3    s> mG>@Sd>%S`S
>20E0 66 21 AA 20 AF 06 04 77 23 10 FC 3E 41 32 AF 20    f!* /..w#.|>A2/
>20F0 ED    m
>S20D6
>20D6 C0 11
>20D7 D3 22
>20D8 64 .*
>V2000,2020,20D0
>2006 C0 11 20D6
>2007 D3 22 20D7
>
>FC.
>*
>X*
>R
       A =00 BC =0000 DE =0000 HL  0000 IX =0000 IY =0000    ...
       A'=00 BC'=0000 DE'=0000 HL' 0000 SP =3FFF PC =0000
>XA
>A=00 99
>XX
>X=2222 8886
>R
       A =99 BC  0000 DE  0000 HL  0000 IX =0000 IY =0000    ...
       A'=00 BC' 0000 DE' 0000 HL' 0000 SP =3FFF PC =0000
```