

205
97



Universidad Nacional Autónoma de México

FACULTAD DE INGENIERIA

**ENLACE DE UNA MICROCOMPUTADORA
CON UNA BURROUGHS B6700**

T E S I S
QUE PARA OBTENER EL TITULO DE :
INGENIERO MECANICO ELECTRICISTA
AREA DE INGENIERIA ELECTRICA Y ELECTRONICA
P R E S E N T A :
MARIO ALBERTO RIVERA PONCE

México, D.F.

1982



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

TESIS CON FALLA DE ORIGEN

TEMARIO.

Tema	Página
1.- Introducción	7
1.1.- Antecedentes	8
1.2.- Objetivos	11
1.3.- Elementos del enlace	12
2.- Computador anfitrión (BURROUGHS B6700)	14
2.0.- Arquitectura básica	16
2.1.- Procesador de Comunicación de datos (DCP)	18
2.1.1.- Adapter Cluster	21
2.1.2.- Line Adapter	23
2.2.- Software de comunicación de datos	24
2.2.1.- Network Definition Language	25
2.2.2.- Data Communications Controller	26
2.2.3.- Código Objeto del DCP	27
2.2.4.- Message Controller System	28
2.2.5.- Descripción de una red	30
2.2.6.- Ejemplo de una red	32
2.3.- Descripción del funcionamiento de la Comunicación de datos en la B6700	48
2.4.- Editor (CANDE)	57
2.5.- Flujo de comandos WFL (Work Flow Language)	59
2.6.- Características de las líneas	60
2.6.1.- Resistencia	62
2.6.2.- Reactancia	64
2.6.3.- Inductancia	64
2.6.4.- Capacitancia	65
2.6.5.- Impedancia	65

2.7.- Modems y otros dispositivos auxiliares	66
2.7.1.- Modos de operación	68
2.7.1.1.- Transmisión Síncrona	68
2.7.1.2.- Transmisión Asíncrona	70
2.7.1.3.- Transmisión Serie	72
2.7.1.4.- Transmisión paralelo	72
2.7.2.- Tipos de canales	77
2.7.3.- Ancho de banda de un canal	78
2.7.4.- Modulación	78
2.7.5.- Dispositivos Auxiliares	79
3.- Microprocesador	80
3.1.- Selección de la microcomputadora adecuada	80
3.1.1.- Consideraciones generales	81
3.1.2.- Datos de Software y Hardware	89
3.1.3.- Consideraciones particulares	93
3.1.4.- Conclusiones	104
3.2.- Descripción del sistema	110
3.3.- Hardware utilizado	115
3.3.1.- Módulos del sistema	115
3.3.1.1.- ZPU (Microprocesador)	115
3.3.1.2.- 16FDC (Controlador de discos)	122
3.3.1.3.- 64KZ (Memoria de acceso aleatorio RAM)	123
3.3.2.- Interfaz de comunicación asíncrona (TU-ART)	125
3.3.2.1.- Velocidades	127
3.3.2.1.- Interrupciones	129
3.4.- Software utilizado	131
3.4.1.- Sistema Operativo	131
3.4.2.- Ensamblador	136
3.5.- Definición del problema	138
3.5.1.- Características del enlace	141

3.5.2.- Limitaciones y posibles mejoras	145
3.5.3.- Diagrama de flujo	148
3.5.4.- Listado	169
3.6.- Desarrollo del programa	209
4.- Conclusiones	212
5.- Bibliografía	219
6.- Apéndices	224
I.- CP/M	224
II.- Bus S-100	232
III.- Z80	246
IV.- Señales EIA RS232-C	259
V.- Protocolo de comunicación	262

CAPITULO PRIMERO
INTRODUCCION

1.- Introducción.

En la Universidad Nacional Autónoma de México, se ha utilizado tradicionalmente el proceso centralizado de computación; es decir, la computadora se encuentra en un recinto específico, y los usuarios que quieran hacer uso de sus servicios, acuden al Centro de Cómputo para la utilización de recursos.

Los servicios en el área de computación que proporciona la Universidad a los diferentes usuarios del Centro de Servicios de Cómputo, son muy variados; estos servicios incluyen: Perforación de tarjetas, recepción de programas, proceso de los mismos y entrega de los resultados; se cuenta así mismo con el servicio de microfilmación. También se cuenta con servicios de teleproceso en las diferentes computadoras con que cuenta el Centro de Cómputo. Este servicio en la actualidad tiene una gran demanda, pues es posible desarrollar programas de aplicación para el funcionamiento interactivo con el computador central. También es posible realizar desarrollo de programas de aplicación a través de el sistema de teleproceso.

Los recursos solicitados por los usuarios cada día son mayores, prueba de esto tenemos los siguientes datos:

1.1.- Antecedentes.

En el año de 1978 se ve la necesidad de proporcionar servicio de teleproceso a los diferentes usuarios de la Universidad. Para esto es necesario homogeneizar y organizar el teleproceso en el Centro de Cómputo; por esto se instala un "rack" de modems y se estandariza la red de teleproceso. Al mismo tiempo se establece el departamento de Electrónica y se funda el departamento de Teleproceso. Se analizan las necesidades de cada dependencia y se les asigna un número de líneas y terminales para poder trabajar y consultar a la computadora central.

La demanda de teleproceso se incrementa y se instalan al mayor número de usuarios posibles, terminales. Con esto pronto se satura la capacidad de puertos de Entrada/Salida de la computadora.

En algunas dependencias de la Universidad, el proporcionar el servicio de teleproceso no es suficiente, pues es necesario procesar un gran número de información y trabajos. Algunos de estos usuarios se encuentran localizados a una gran distancia de la Ciudad Universitaria, por lo que resulta difícil y complicado transportar las tarjetas, cintas y listados de un lugar a otro. En 1979 se instalan los RJE's (Remote Job Entry) ó Entrada Remota de Trabajos. Esto se realiza con un minicomputador Eclipse fabricado por Data General; el costo de estos equipos es muy elevado pero a través de ellos es posible el cargar trabajos a la computadora central. Es decir, se cuenta con un "mini-centro" remoto en el cual se pueden leer las tarjetas correspondientes a los trabajos de los usuarios, realizar archivos y colas para que los usuarios introduzcan los trabajos. Cuando es posible la transmisión de información al computador central, se transmite a través de una línea

telefónica para su proceso en el computador central. Una vez procesada la información, el computador central transmite al "mini-centro" remoto las impresiones ocasionadas por los trabajos previamente introducidos. Con este tipo de sistemas es posible la conexión de terminales de teleproceso a través del minicomputador.

En el año de 1981 se instalan los concentradores, los cuales son dispositivos electrónicos, por medio de los cuales es posible la conexión de varias terminales a través de una sola línea física. Es decir, en una línea física se tienen hasta cinco líneas lógicas. Este concentrador fue desarrollado en forma conjunta por el Centro de Instrumentos y el Centro de Servicios de Cómputo. Para ello se utilizó el microprocesador M6800. El diseño y la construcción de este equipo se realizó totalmente en la Universidad.

A mediados del año 1981 se plantea la posibilidad de conectar terminales "Super Inteligentes" a la red de teleproceso. Para esto se opta por la utilización de microcomputadores, con el fin de que la atención a la línea del computador central, la realice el microprocesador de éste y no el de la terminal. Con esto se tiene un gran número de ventajas que normalmente no se tienen.

El rápido desarrollo de la tecnología ha permitido desarrollar microprocesadores por menos de 10 dólares, así mismo los sistemas microcomputadores han sufrido un decremento considerable en sus precios, de tal manera que es posible adquirir equipos de éste tipo a un costo bajísimo (desde 100 dólares en una configuración básica) a comparación de los grandes equipos y sistemas tradicionales de cómputo.

En base a las necesidades del tiempo de procesamiento, así como tiempo de Entrada/Salida, la reducción de los precios de los equipos de microcomputadoras, permitirá y requerirá la compra de varios de estos sistemas, con una configuración tal, que permita principiar a trabajar con los

periféricos mínimos necesarios, y posteriormente poder conectar más periféricos para una utilización más elevada de estos sistemas.

Es mejor tener varios equipos microcomputadores inactivos, que un gran equipo de cómputo inactivo. Pues las estadísticas revelan que la mayor utilización de un equipo de cómputo es en horas hábiles, quedando todo el tiempo restante casi sin utilizarse. En el caso de un gran equipo de cómputo, es necesario tener equipos auxiliares tales como: suministro de energía ininterrumpida, equipo de aire acondicionado. Estos equipos elevan los costos iniciales, así como de mantenimiento y el tener un sistema de este tipo sin utilizarse, es un desperdicio de recursos.

Con los sistemas microcomputadores se pretende que el mayor trabajo se realice en forma local y el trabajo mayor se transfiera al equipo central.

1.2.- Objetivos.

Con la utilización de microcomputadoras en la red de teleproceso, se pretende que los usuarios realicen el máximo posible de trabajo en forma local; es decir, la edición de archivos de datos y programas de aplicación se pueden hacer, utilizando los editores de las microcomputadoras y estos archivos se transmitirán a la computadora central para su compilación y ejecución.

Para los usuarios donde se requiera la captura de datos en tiempo real, será posible monitorear cualquier tipo de proceso, muestrear los datos necesarios, guardarlos en el sistema microcomputador, realizar algún tipo de proceso con los datos obtenidos, (como puede ser: codificar, modificar, etc) y finalmente, transmitirlos al computador central para su proceso masivo.

La perforación de tarjetas de datos y programas, puede ser substituida por sistemas microcomputadores, en los cuales se simule una perforadora de tarjetas, con sus rutinas de verificación, repetición de campos, etc. En otro computador (o uno de los destinados para la "perforación") se leerán estos archivos conteniendo imágenes de tarjetas y serán transmitidos al computador central para su procesamiento habitual de tarjetas perforadas.

1.3.- Elementos de enlace.

En conclusión, los elementos de enlace necesarios para este proyecto son:

Las computadoras BURROUGHS que posee la Universidad y varios microcomputadores, que serán adquiridos.

En la BURROUGHS se cuenta con el "hardware" (dispositivos electrónicos, eléctricos, magnéticos y mecánicos) necesario para la realización de este proyecto. Así mismo se cuenta con el software (Programas y sistemas) para la comunicación de datos entre los sistemas mencionados. En el Capítulo Segundo se describen ambos aspectos.

El microcomputador que se utilizará deberá ser seleccionado de tal forma que se obtenga el mejor servicio posible de este equipo y tratar de estandarizar este tipo de sistemas en la Universidad. El Capítulo Tercero en su primera parte trata de esta selección.

La Segunda y Tercera parte de este capítulo se refieren a las características de hardware del equipo seleccionado. En la Cuarta parte del mismo capítulo se describen las características de software de dicho equipo.

En la última parte del Tercer Capítulo se describe la solución del problema, así como de su solución específica. Se incluye un listado del programa desarrollado para solucionar el problema del enlace de BURROUGHS con un microprocesador.

En el Capítulo Cuarto se obtienen las conclusiones de el presente trabajo de tesis.

La bibliografía y obras de consulta están incluidas en el Capítulo Quinto.

Por último, en el capítulo sexto se incluyen los apéndices necesarios.

Nota: El desarrollo tecnológico en el área de computación se realiza principalmente en los Estados Unidos de Norte América, debido a esto, existen un sinnúmero de tecnicismos que resultan imposibles de traducir o castellanizar. Trataremos de utilizar al mínimo los Anglicismos, pero cuando no exista un equivalente en Español, recurriremos a la palabra original en Inglés.

C A P I T U L O S E G U N D O
C O M P U T A D O R A N F I T R I O N B 6 7 0 0

2.- Computador Anfitrión (BURROUGHS B6700).

La Burroughs B6700 es un sistema de información de mediana a alta velocidad. Las siguientes, son algunas de las características sobresalientes de este sistema:

- 1.- Circuitería Monolítica.
- 2.- Memoria expandible a 1,048,576 palabras.
- 3.- Tiempo de ciclo de memoria de 1.2 microsegundos, 1.5 microsegundos y 500 nanosegundos.
- 4.- Configuración periférica expandible a 256 unidades.
- 5.- Capacidad de hasta tres procesadores de Entrada/Salida permitiendo con esto hasta 36 operaciones simultáneas de E/S.
- 6.- Software para comunicación de datos para cómputo remoto y manipulación de archivos.
- 7.- Almacenamiento en disco de más de 36×10^6 a la (12) bytes (en caracteres de 8 bits).

Esta es una tecnología atrazada, pues se lanzó al mercado a finales de 1970, y se estableció en la Universidad en 1972. En la actualidad la Universidad cuenta con dos de estos equipos, que han estado en operación continua por casi

10 años.

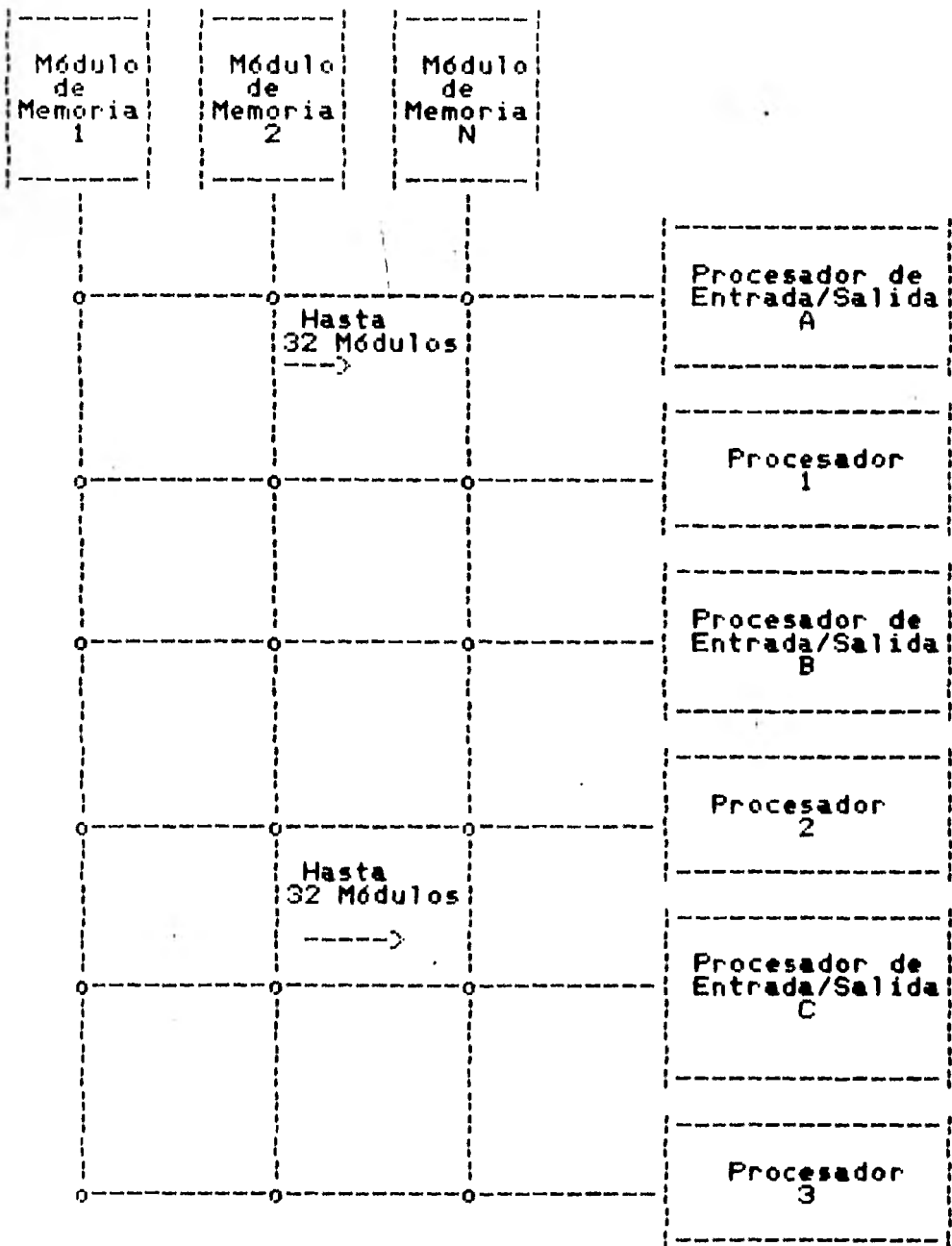
La tecnología actual ha desarrollado la utilización de más de un procesador en los sistemas de cómputo.

Generalmente se tiene uno o varios procesadores encargados de atender a los periféricos, tales como: Impresoras, lectoras y perforadoras de tarjetas, unidades de disco y unidades de cinta. A este tipo de procesadores se les llama procesadores de Entrada/Salida o también "Back-End Storage Processors". Con este tipo de procesadores se han realizado inclusive redes de almacenamiento (ver referencia 2.6).

También se cuenta con procesadores destinados a la comunicación de datos; es decir, las líneas físicas de las diferentes terminales llegan a este procesador, el cual se encarga de realizar la comunicación con los diferentes usuarios de teleproceso. A este tipo de procesadores se les llama comúnmente "Front-End Processor". Todas las máquinas de gran tamaño tienen uno o varios de estos.

Se decidió realizar la comunicación entre esta máquina y una microcomputadora, debido a que las principales aplicaciones serán en el área administrativa, que está situada en estos equipos. Las arquitecturas de estos sistemas permiten que el software desarrollado para esta máquina funcione para sistemas mucho más recientes.

2.0.- Arquitectura básica.



Representación de la Configuración de la B6700.

La configuración de este sistema puede ser la siguiente:

- Procesador central	1 - 3
- Procesador de E/S	1 - 3
- Procesadores de	
Comunicación de datos	1 - 4
- Módulos de memoria	1 - 32
- Unidades de disco	1 - 20
- Unidades de cinta	1 - 16
- Unidades de consola	1 - 4
- Lectoras de tarjetas	1 - 4
- Impresoras	1 - 4
- Terminales	1 - 768

2.1.- Procesador de comunicación de datos (DCP).

El procesador de comunicación de datos (DCP), es una minicomputadora programable de propósito especial, puede (si así se desea) tener su propia memoria. Contiene los registros necesarios y la lógica requerida para ejecutar todas las funciones básicas asociadas con la recepción y transmisión de datos. Opera en forma independiente del equipo; es decir, opera en forma asíncrona del sistema central (B6700) y hace las veces de interfaz de comunicación entre el sistema central y los concentradores de puertos (llamados Adapter Cluster); los cuales pueden ser de 1 a 16, según sean los requerimientos).

Las funciones del DCP son las siguientes:

- a).- Manejo de estaciones o terminales remotas que se encuentran asociadas a cada una de las líneas de comunicación. Controlar el formato de los datos que son transmitidos y recibidos.
- b).- Acumular los caracteres que se reciben de una terminal (a través del Adapter Cluster) en conjuntos de palabras de 6 caracteres formando mensajes que serán transmitidos a la memoria principal, la cual está localizada en el sistema central.
- c).- Extrae desde el sistema central mensajes, palabra por palabra, para transmitir carácter por carácter hacia el Adapter Cluster, el cual

lo transmitirá hacia una terminal.

- d).- Es el encargado de traducir los códigos que se requieran, por ejemplo: del código ASCII (el que normalmente utilizan las terminales y teletipos) al código EBCDIC (el que maneja internamente la BURROUGHS).
- e).- Realiza las validaciones de paridad.
- f).- Ejecuta rutinas básicas de recuperación de errores, por ejemplo pérdida de un bit; si el programador lo especifica.
- g).- Obedece cuando es posible a varias peticiones del sistema central, tales como, llamar por teléfono a una línea específica y participar en la reconfiguración dinámica de la red.

Los elementos básicos del DCP son:

- i).- "Scratch-pad-memory".

Consta de 8 palabras de 52 bits cada una. Esta es una memoria de circuitos integrados con un tiempo de lectura de 40 nanosegundos y un tiempo de escritura de 50 nanosegundos.

- ii).- Memoria local.

Esta memoria es modular del tipo de núcleos magnéticos con un ciclo de 650 nanosegundos, cada módulo consiste de 4096 palabras de 52 bits cada una.

iii).- Registros varios.

Estos registros se clasifican en dos categorías: manejadores de datos e información de control y manejadores de instrucciones. En total son 14, y estos son:

AA	AC
AI	D
Y	X
CF	MAO
MA1	MA2
1A	BA
IR	W

Registros del DCP.

2.1.1.- Adapter Cluster.

Este sirve como interfaz entre el DCP y los Line Adapters. Un DCP puede tener hasta 16 Adapter Cluster y cada uno de ellos sirve como un multiplexor a 16 líneas ó Line Adapters.

El Adapter Cluster recibe de los Line Adapters los bits que llegan y los acumula para formar caracteres, los cuales son transmitidos al DCP. En el proceso contrario, descompone los caracteres en bits, para ser transmitidos a los Line Adapters.

Los elementos básicos de éste son:

i).- "Scratch-pad-memory".

Es una memoria de circuitos integrados (IC) con un tiempo de acceso de 40 nanosegundos en la lectura y 50 nanosegundos en la escritura. Consiste de 16 palabras de 56 bits cada una; una palabra para cada uno de los Line Adapters que se le pueden conectar.

ii).- BAR.

El "Buffer Associative Register" es el registro más importante del Adapter Cluster y lo utilizan los Line Adapters en tiempo compartido; consiste de varios campos donde almacena temporalmente información de: datos, control y tiempo.

El registro "BAR" es utilizado para almacenar los resultados de las operaciones lógicas. El reloj de tiempo real es un reloj con una frecuencia de 5 MHz. Todo el procesamiento de la información en el Adapter Cluster ocurre cuando extrae una palabra de su scratch memory y la pone en el "BAR"; simultáneamente, reemplaza la palabra extraída previamente ; efectúa la transmisión de caracteres y bits entre el DCP y los Line Adapters relativos a la palabra puesta en el "BAR"; de esta forma el Adapter Cluster atiende a su memoria cada 3.2 microsegundos.

2.1.2.- Line Adapter.

Los Line Adapters son dispositivos electrónicos que proveen una interfaz entre el Adapter Cluster y el dispositivo que se conectará a él. Por ejemplo: modems, line-drivers, líneas directas. Básicamente ajusta los cambios de niveles de voltaje entre ambos dispositivos, y es donde el dispositivo externo y el Adapter Cluster se sincronizan entre sí.

El Line Adapter puede ser de varios tipos, dependiendo de las velocidades de transmisión y de la forma de conectar a éste los dispositivos; algunos de estos dispositivos son:

- i).- Cable directo y velocidades de 150 a 9600 Bauds.
- ii).- Conexión de cable a través de un módem a velocidades de 150 a 2400 Bauds.
- iii).- Unidad automática de llamado "ACU".
- iv).- Línea de respuesta de audio, "Audio Response Line".
- v).- Entrada de teléfono tipo "touch tone".

Si una terminal es conectada vía módem a la línea, se le denomina a este tipo de conexión "línea no directa" o terminal remota.

2.2.- Software de comunicación de datos.

Los componentes del software para comunicación de datos, son los siguientes:

- a).- Programa fuente en NDL (Network Definition Language).
- b).- Compilador de NDL.
- c).- DCC (Data Communications Controller).
- d).- Código objeto del DCP.
- e).- Sistema controlador de mensajes (MCS).
- f).- Compilador de DCALGOL (Data Communications Algol).

2.2.1.- Network Definition Language.

El programa fuente en NDL, es un programa en el cual se describe física, funcional y lógicamente la red de comunicación de datos; o sea, los atributos físicos y el manejo de las líneas, así como el procedimiento para el manejo de datos. Está escrito en un lenguaje llamado NDL y se compila con el compilador del mismo nombre, para generar una base de datos tabular. En esta base de datos están las características físicas y lógicas de la red y un conjunto de algoritmos de manejo de las capacidades del sistema de comunicación de datos. Cuando un usuario define exactamente cuales son las características de la red de comunicación de datos, se utiliza el NDL para describir su naturaleza exacta, por ejemplo; el número y el tipo de dispositivos remotos que son conectados a la red, el número y el tipo de las líneas de transmisión disponibles.

El compilador de NDL es un programa que compila el código fuente de NDL y como resultado de ésta compilación produce lo siguiente:

- a).- El código objeto para el DCP (DCPCODE).
- b).- Tablas usadas para el Sistema Central y el DCP NIF (Network Information File).

2.2.2.- Data Communication Controller.

El DCC es la parte del sistema operativo que se encarga de la comunicación de datos. Esta es la interfaz básica entre el DCP y el sistema principal. Existe como un subconjunto básico del MCP (Master Control Program), y opera independientemente en proceso o en stack. En un proceso por cada DCP activo.

Cada estación está controlada por un sistema controlador de mensajes (MCS) a la vez. Varias estaciones pueden ser controladas por diferentes MCS's. Todos los movimientos de información entre el sistema central y el DCP son controlados por el DCC. El DCC actúa como intermediario entre los MCS's y los programas de aplicación.

El DCC también es el encargado de checar la validez de las instrucciones pasadas por los MCS's a los DCP's.

2.2.3.- Código objeto del DCP.

El código objeto del DCP es un sistema operativo para el DCP, las funciones de éste dependen del programa fuente NDL; algunas de estas funciones son:

- a).- Operaciones de multiplexeos de líneas.
- b).- Multiplexeos de las peticiones del sistema.
- c).- Control de líneas.
- d).- Recolección y distribución de caracteres.
- e).- Detección de errores.

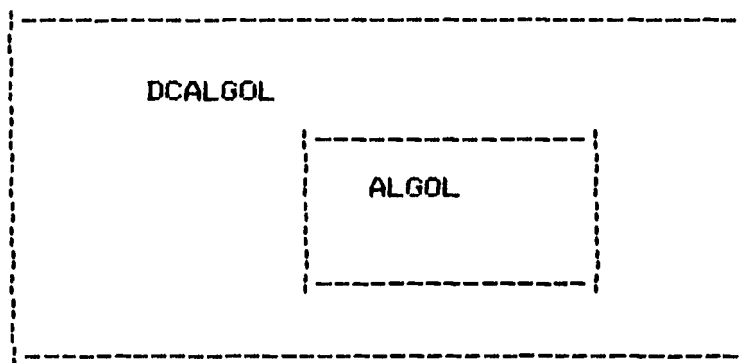
2.2.4.- Message Controller System.

Un MCS es un programa de propósito especial escrito en DCALGOL (Data Communications ALGOL), que puede ser proporcionado por el fabricante (por ejemplo: SISTEM/CANDE, SYSTEM/RJE Y SYSTEM/DIAGNOSTICSMCS) , o bien puede ser escrito por los usuarios. Un MCS, como su nombre lo implica, controla el flujo de mensajes de comunicación de datos entre la terminal y el sistema principal.

La información de el DCP, tal como la entrada de una terminal e información del estado, es dirigida al MCS a través del DCC. Los mensajes del MCS al DCP, tales como salida de una terminal de la red de teleproceso, o cambios de red, son realizados por el MCS invocando una función intrínseca de la máquina llamada "DCWRITE". Cada estación que está definida por el programa objeto en NDL, deberá tener uno y solo uno de los MCS que lo controle.

Un MCS puede controlar líneas a las cuales está asignado, en las cuales hay al menos una estación que controla el MCS particular.

Este control consiste entre otras cosas, en poner la estación lista para funcionar, y asignar a la estación un archivo de trabajo.



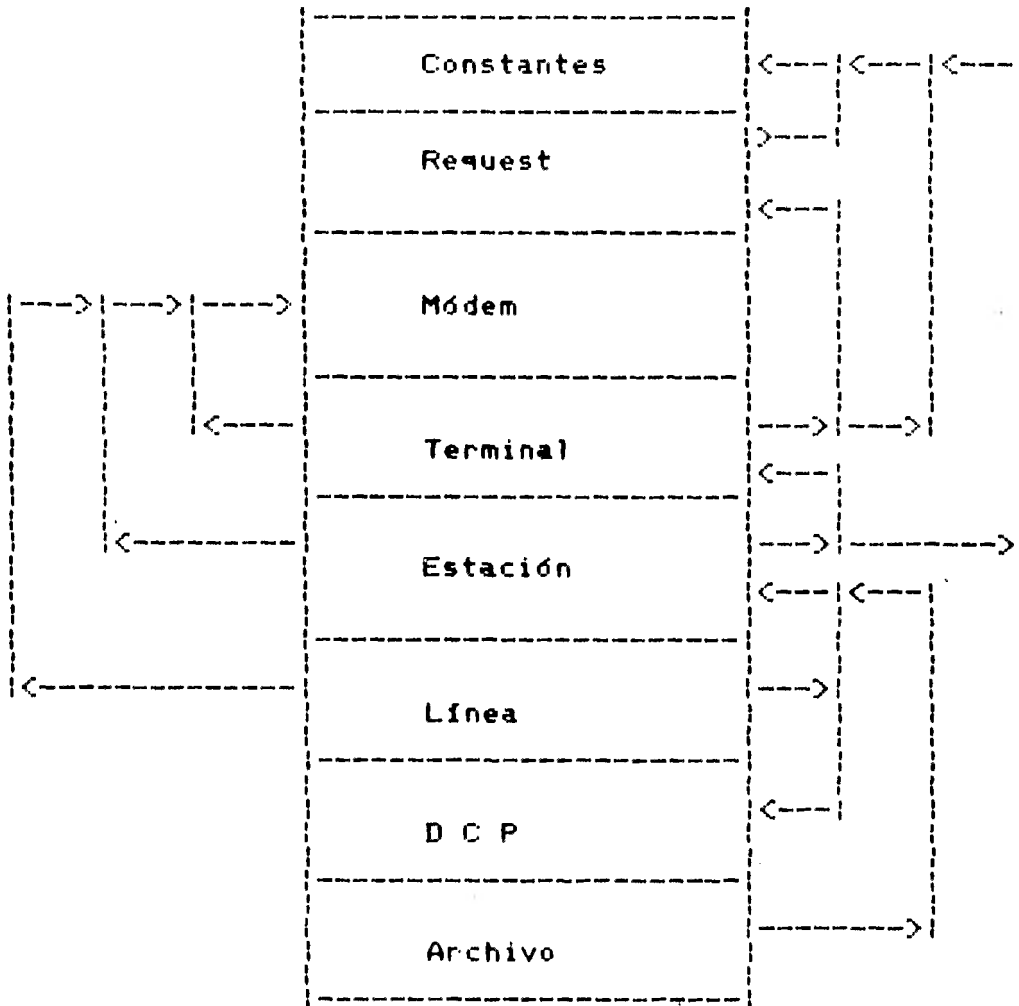
E1 DCALGOL y e1 ALGOL.

2.2.5.- Descripción de una red de comunicación de datos.

Para describir una red de comunicación de datos, se escribe un programa en NDL, en donde se tiene la siguiente estructura:

- a).- Constantes.
- b).- Request.
- c).- Módem.
- d).- Terminal.
- e).- Estación.
- f).- Línea.
- g).- DCP
- h).- Archivo.

Existen listas lógicas entre las diferentes secciones de la red; estas relaciones son:



* referencia 2.2 y 2.5

2.2.6.- Ejemplo de una red.

A continuación se incluye un listado del resultado de una compilación de la red de comunicación de datos utilizando el compilador NDL.


```
6900 TWENTYSL: C 20037000
7000 SCR = 20. C 20038000
7100 GO ZEROL. C 20039000
7200 TENS L: C 20040000
7300 SCR = 10. C 20041000
7400 ZEROL: C 20042000
7500 RECEIVE (100 MILLI)[0]. C 20043000
7600 CHAR = CHAR AND 4"OF". C 20044000
7700 SCR = SCR + CHAR. C 20045000
7800 IF SCR > 3 THEN % TRANS # GAP FROM 3 TO 11 C 20046000
7900 SCR = SCR -7. C 20047000
8000 % SCR IS NOW STATION # C 20048000
8100 END#, C 20049000
8200 % C 20050000
8300 % WRITETELETYPE DEBLANKER C 20051000
8400 % C 20052000
8500 % C 20053000
8600 % CONCENTRATOR CARRIAGE CONTROL C 20054000
8700 % C 20055000
8800 CARRIAGECONTROL = C 20056000
8900 BEGIN C 20057000
9000 CHAR = 0. C 20058000
9100 IF BLOCK THEN C 20059000
9200 CHAR[3] = TRUE. C 20060000
9300 IF PAGE THEN C 20061000
9400 CHAR[2] = TRUE. C 20062000
9500 IF CARRIAGE THEN C 20063000
9600 CHAR[1] = TRUE. C 20064000
9700 IF LINEFEED THEN C 20065000
9800 CHAR[0] = TRUE. C 20066000
9900 % MAKE AN ASCII EQUALVALENT CHAR C 20067000
10000 % THAT ISNT A SPECIAL CHARACTER C 20068000
10100 % C 20069000
10200 IF SPACE THEN C 20070000
10300 IF SKIPCONTROL = 0 THEN C 20071000
10400 CHAR = CHAR AND 4"FC". % WRITE STOP== SPACE[0] C 20072000
10500 IF CHAR = 0 THEN C 20073000
10600 CHAR = "@" ELSE C 20074000
10700 IF CHAR GTR 9 THEN C 20075000
10800 CHAR = CHAR + 4"C7" C 20076000
10900 ELSE C 20077000
11000 CHAR = CHAR + 4"CO". C 20078000
11100 TRANSMIT CHAR. C 20079000
11200 END#, C 20080000
11300 % C 20081000
11400 % WRITE RJE DEBLANKER C 20082000
11500 % C 20083000
11600 RJEDELETEBLANK(BLNKCNT, TEMPCHAR) = C 20084000
11700 BEGIN C 20085000
11800 BLNKCNT = 0. C 20086000
11900 FETCHLOOP: C 20087000
12000 FETCH[ENDOFTEXTL]. C 20088000
12100 IF CHAR = " " THEN C 20089000
12200 BEGIN C 20090000
12300 BLNKCNT = BLNKCNT + 1. C 20091000
12400 DELAY(250 MICRO). C 20092000
12500 GO FETCHLOOP. C 20093000
12600 END C 20094000
12700 ELSE C 20095000
12800 TEMPCHAR = CHAR. C 20096000
12900 TRANSMITL: C 20097000
13000 IF BLNKCNT = 0 THEN C 20098000
13100 BEGIN C 20099000
13200 CHAR = TEMPCHAR. C 20100000
13300 IF CHAR = ETX THEN C 20101000
13400 CHAR = QM. C 20102000
13500 IF CHAR = SYN THEN C 20103000
13600 CHAR = QM. C 20104000
```

```
13700          TRANSMIT.                                C 20105000
13800          END                                     C 20106000
13900          ELSE BEGIN                             C 20107000
14000             BLNKCNT = BLNKCNT-1.                C 20108000
14100             TRANSMIT " ".                       C 20109000
14200             GO TRANSMITL.                        C 20110000
14300          END.                                    C 20111000
14400          GO TO FETCHLOOP.                        C 20112000
14500          ENDOFTEXTL:                             C 20113000
14600          IF BLNKCNT GTR 0 THEN                   C 20114000
14700             IF NOT CARRIAGE THEN                 C 20115000
14800                 BEGIN                             C 20116000
14900                     TEMPCHAR = " ".              C 20117000
15000                     BLNKCNT = BLNKCNT-1.         C 20118000
15100                     GO TRANSMITL.                C 20119000
15200                 END.                              C 20120000
15300          END#.                                    C 20121000
15400
15500 %***** RJE *****                                C 30000000
15600 %          TOG [0] BAD RESPONSE                     C 30001000
15700 %          TOG [1] ESTABLISHED                     C 30002000
15800 %          TOG [2] NEED TO ACK                     C 30003000
15900 %          TOG [3] NEED TO NAK                     C 30004000
16000 %          TOG [4] SENT MESSAGE                    C 30005000
16100 %          TOG [5] HAVE BEEN ACKED                 C 30006000
16200 %          TOG [6] LOWERF LAST SENT.               C 30007000
16300 %          TOG [7] NAK RESPONSE.                   C 30008000
16400 %          TALLY[2][0] WE CAME FROM LINECONTROL TO TRANSMIT REQ. C 30009000
16500 %          TALLY[2][1] TRANS# REQ LAST TRANS# RECEIVED C 30010000
16600 %          TALLY[2][2] NEED TO RECEIVE ADDRESS.    C 30011000
16700 %          TALLY[2][3] BAD ADDRESS CHARACTERS      C 30012000
16800 %          TALLY[2][7] UCSD CONCENTRATOR TERMINAL  C 30013000
16900 %          TALLY[2][4] ESTABLISH UCSD TERMINAL TYPES C 30014000
17000 %          TALLY[2][5] WAITING, TRANSMISSIONS HAVE BEEN SUSPENDE C 30015000
17100 %          C 30016000
17200 %          LINE(TALLY[0]) LAST STATION TO TRANSMIT. C 30017000
17300 %          LINE(TALLY[1]) CNR TRANSMIT              C 30018000
17400 %          LINE(TOG[0]) REQUEST NOT INITIATED FROM LINE CONTROL C 30019000
17500 %          LINE(TOG[1]) TRANS# SENT=CNR RECEIVED    C 30020000
17600 %          C 30021000
17700 %*****
17800          CONTROL CONCENTRATE:                       C 30023000 CONCENTRATE
17900          ERROR[0] =                                 C 30044000 CONCENTRATE
18000          TIMEOUT:20                                C 30045000 CONCENTRATE
18100          ,PARITY:20                                 C 30046000 CONCENTRATE
18200          ,LOSSOF CARRIER:20                       C 30047000 CONCENTRATE
18300          ,STOPBIT:20                               C 30048000 CONCENTRATE
18400          ,BUFOVFL:20                               C 30049000 CONCENTRATE
18500          .                                          C 30050000 CONCENTRATE
18600          CAMEFROMCONTROL = FALSE.                  C 30051000 CONCENTRATE
18700          IF BADINITIATE THEN                       C 30052000 CONCENTRATE
18800              BEGIN                                 C 30053000 CONCENTRATE
18900                  BADINITIATE = FALSE.             C 30054000 CONCENTRATE
19000                  STATION = 0.                      C 30055000 CONCENTRATE
19100              END.                                  1 C 30056000 CONCENTRATE
19200          CHAR = TOGS.                               C 30057000 CONCENTRATE
19300          IF NOT CHAR(ESTABLISHED) THEN             C 30058000 CONCENTRATE
19400              BEGIN                                 1 C 30059000 CONCENTRATE
19500                  STATION=0.                        C 30060000 CONCENTRATE
19600                  INITIATE ENABLEINPUT.             C 30061000 CONCENTRATE
19700                  IDLE.                             C 30062000 CONCENTRATE
19800              END.                                  1 C 30063000 CONCENTRATE
19900 %          IF NOT STATION(VALID) THEN GO TO 100.    C 30064000 CONCENTRATE
20000 %          IF NOT STATION(READY) THEN GO TO 100.    C 30065000 CONCENTRATE
20100 %          IF ESTABLISHTERM THEN % MARK ALL STATIONS AS C 30066000 CONCENTRATE
20200 %          BEGIN                                     1 C 30067000 CONCENTRATE
20300 %             ESTABLISHL:                             C 30068000 CONCENTRATE
20400 %             IF STATION NE MAXSTATIONS THEN         C 30069000 CONCENTRATE
```

```
20500          BEGIN                                2 C 30070000 CONCENTRATE
20600          DELAY(200 MICRO).                    C 30071000 CONCENTRATE
20700 %        IF TERMINAL = RJEPTAMOD THEN          C 30072000 CONCENTRATE
20800 %        UCSDTERMINAL = TRUE ELSE             C 30073000 CONCENTRATE
20900 %        IF TERMINAL = RJEAMOD OR TERMINAL = RJEEMOD THEN C 30074000 CONCENTRATE
21000 %        UCSDTERMINAL = TRUE ELSE             C 30075000 CONCENTRATE
21100 %        IF TERMINAL = RJEASYTEK4006 THEN     C 30076000 CONCENTRATE
21200 %        UCSDTERMINAL = TRUE ELSE             C 30077000 CONCENTRATE
21300 %        IF TERMINAL = RJEASYADM1 THEN        C 30078000 CONCENTRATE
21400 %        UCSDTERMINAL = TRUE ELSE             C 30079000 CONCENTRATE
21500 %        IF TERMINAL = RJEASYHAZELTINE THEN   C 30080000 CONCENTRATE
21600 %        UCSDTERMINAL = TRUE ELSE             C 30081000 CONCENTRATE
21700 %        IF TERMINAL = RJEASYTI THEN          C 30082000 CONCENTRATE
21800 %        UCSDTERMINAL = TRUE                 C 30083000 CONCENTRATE
21900 %        ELSE                                  C 30084000 CONCENTRATE
22000 %        UCSDTERMINAL = FALSE.                 C 30085000 CONCENTRATE
22100 %        FREQNTR = 0.                          C 30086000 CONCENTRATE
22200 %        WAITING = FALSE. % RJE RESTART RESETS THIS C 30087000 CONCENTRATE
22300 %        STATION = STATION +1.                 C 30088000 CONCENTRATE
22400 %        GO ESTABLISHL.                        C 30089000 CONCENTRATE
22500 %        END.                                  2 C 30090000 CONCENTRATE
22600 %        STATION = 0.                          C 30091000 CONCENTRATE
22700 %        ESTABLISHTERM = FALSE.                C 30092000 CONCENTRATE
22800 %        END.                                  1 C 30093000 CONCENTRATE
22900 %        IF GETADDRESS THEN                    C 30094000 CONCENTRATE
23000 %        BEGIN                                  1 C 30095000 CONCENTRATE
23100 % RECEIVE ADDRESS CHARACTERS AND SWITCH STATION C 30096000 CONCENTRATE
23200 %        GETADDRESS=FALSE.                     C 30097000 CONCENTRATE
23300 %        BCC=TALLY[0].                          C 30098000 CONCENTRATE
23400 %        RECEIVEADDRESS(TALLY[1]).              2 C 30099000 CONCENTRATE
23500 %        CHAR=TOGS.                              C 30100000 CONCENTRATE
23600 %        CHAR[NEEDACK] = NEWTRANSM. % SAVE      C 30101000 CONCENTRATE
23700 %        STATION=TALLY[1]. % THIS BIT          C 30102000 CONCENTRATE
23800 %        TALLY[0]=BCC.                          C 30103000 CONCENTRATE
23900 %        GETADDRESS=TRUE.                      C 30104000 CONCENTRATE
24000 %        BADADDRESS=FALSE.                    C 30105000 CONCENTRATE
24100 %        TOGS=CHAR.                            C 30106000 CONCENTRATE
24200 %        NEWTRANSM = CHAR[NEEDACK]. % RESTORE  C 30107000 CONCENTRATE
24300 %        TOG[1] = TRUE. % THIS BIT             C 30108000 CONCENTRATE
24400 %        INITIATE ENABLEINPUT.                  C 30109000 CONCENTRATE
24500 %        30: CHAR=TOGS.                          C 30110000 CONCENTRATE
24600 %        GO TO 40.                              C 30111000 CONCENTRATE
24700 %        20:                                     C 30112000 CONCENTRATE
24800 %        % NOW FIND A GOOD STATION AND DUMP THE ERROR C 30113000 CONCENTRATE
24900 %        CHAR=TOGS. % ON HIM.....              C 30114000 CONCENTRATE
25000 %        CHAR[NEEDNAK]=TRUE.                   C 30115000 CONCENTRATE
25100 %        40:                                     C 30116000 CONCENTRATE
25200 %        STATION=0.                              C 30117000 CONCENTRATE
25300 %        GETADDRESS=TRUE.                       C 30118000 CONCENTRATE
25400 %        BADADDRESS=TRUE.                      C 30119000 CONCENTRATE
25500 %        GO TO 61.                              C 30120000 CONCENTRATE
25600 %        END.                                  1 C 30121000 CONCENTRATE
25700 %        % WE GOT HERE FROM SOMEWHERE ELSE...   C 30122000 CONCENTRATE
25800 %        IF LASTSTATION=255 THEN                 C 30123000 CONCENTRATE
25900 %        BEGIN                                  1 C 30124000 CONCENTRATE
26000 %        LASTSTATION=STATION.                    C 30125000 CONCENTRATE
26100 %        60:                                     C 30126000 CONCENTRATE
26200 %        % FIND AN INPUTTER                      C 30127000 CONCENTRATE
26300 %        STATION=0.                              C 30128000 CONCENTRATE
26400 %        61:                                     C 30129000 CONCENTRATE
26500 %        TALLY[0]=BCC.                          C 30130000 CONCENTRATE
26600 %        TOGS=CHAR.                              C 30131000 CONCENTRATE
26700 %        INITIATE ENABLEINPUT.                   C 30132000 CONCENTRATE
26800 %        IDLE.                                   C 30133000 CONCENTRATE
26900 %        END.                                  1 C 30134000 CONCENTRATE
27000 %        IF WAITING THEN % IF STA 0 WE ARE FROM RECEIVING C 30135000 CONCENTRATE
27100 %        IF LASTSTATION NE 0 THEN % SUS/ACK/NAK C 30136000 CONCENTRATE
27200 %        IF LASTSTATION NE 0 THEN % SUS/ACK/NAK C 30137000 CONCENTRATE
```

```
27300 IF STATION = 0 THEN
27400 BEGIN
27500 WAITING = FALSE.
27600 STATION = LASTSTATION.
27700 WAITING = TRUE.
27800 FREQCNTR = 60. %OVERRIDE WAIT IN ABOUT 3 MINUTES
27900 END. % GOOD OR BAD MESSAGE SENT/TO RESEND
28000
28100 75:
28200 IF CHAR[SENTMESSAGE] THEN
28300 IF CHAR[BADRESPONSE] THEN
28400 BEGIN
28500 STATION=LASTSTATION.
28600 FREQCNTR = 0. % PROCESS BREAK NOW
28700 GO TO 125.
28800 END.
28900 ELSE
29000 IF CHAR[BEENACKED] THEN
29100 BEGIN
29200 %AHAH-GOOD MESSAGE SENT)
29300 STATION=LASTSTATION.
29400 TOGS=CHAR.
29500 CAMEFROMCONTROL=TRUE.
29600 INITIATE REQUEST.
29700 CHAR[SENTMESSAGE]=FALSE.
29800 IF TOG[SENTLOWERF] THEN
29900 TOG[SENTLOWERF]=FALSE ELSE
30000 TOG[SENTLOWERF]=TRUE.
30100 END.
30200 % NEED SOMETHING TO DO....
30300
30400 100: IF LASTSTATION GEQ MAXSTATIONS THEN
30500 LASTSTATION=MAXSTATIONS-1.
30600 STATION=LASTSTATION.
30700 IF CHAR[SENTMESSAGE] THEN
30800 IF WAITING THEN
30900 BEGIN
31000 CHAR[SENTMESSAGE] = FALSE.
31100 CHAR[BADRESPONSE] = FALSE.
31200 CHAR[BEENACKED] = FALSE.
31300 END ELSE
31400 IF NOT CHAR[BEENACKED] THEN
31500 CHAR[BADRESPONSE]=TRUE.
31600 IF CHAR[BADRESPONSE] THEN
31700 BEGIN
31800 TOGS=CHAR.
31900 CAMEFROMCONTROL=TRUE.
32000 INITIATE REQUEST.
32100 END.
32200 END.
32300 LINE(QUEUED) = TRUE.
32400
32500 101: IF STATION>0 THEN
32600 BEGIN
32700 DELAY(1000 MICRO).
32800 STATION=STATION-1.
32900
33000 125:
33100 % IF STATION(VALID) THEN
33200 IF STATION(READY) THEN
33300 BEGIN
33400 IF STATION(QUEUED) THEN
33500 IF FREQCNTR > 0 THEN
33600 BEGIN
33700 FREQCNTR = FREQCNTR -1.
33800 IF NOT WAITING THEN
33900 BEGIN
34000 IF LINE(QUEUED) THEN
IF STATION=LASTSTATION THEN
```

```
34100 LINE(QUEUED) = FALSE C 30206000 CONCENTRATE
34200 ELSE C 30207000 CONCENTRATE
34300 ELSE C 30208000 CONCENTRATE
34400 GO 110. C 30209000 CONCENTRATE
34500 GO TO 101. C 30210000 CONCENTRATE
34600 END. 4 C 30211000 CONCENTRATE
34700 END 3 C 30212000 CONCENTRATE
34800 ELSE C 30213000 CONCENTRATE
34900 BEGIN 3 C 30214000 CONCENTRATE
35000 110: C 30215000 CONCENTRATE
35100 IF NOT LINE(QUEUED) THEN C 30216000 CONCENTRATE
35200 LASTSTATION=STATION. C 30217000 CONCENTRATE
35300 CHAR(BADRESPONSE)=FALSE. C 30220000 CONCENTRATE
35400 TOGS=CHAR. C 30221000 CONCENTRATE
35500 CAMEFROMCONTROL=TRUE. C 30222000 CONCENTRATE
35600 WAITING=FALSE. C 30223000 CONCENTRATE
35700 INITIATE REQUEST. C 30224000 CONCENTRATE
35800 END. 3 C 30225000 CONCENTRATE
35900 IF STATION NE LASTSTATION THEN GO TO 101. C 30226000 CONCENTRATE
36000 IF LINE(QUEUED) THEN C 30227000 CONCENTRATE
36100 BEGIN 3 C 30228000 CONCENTRATE
36200 LINE(QUEUED) = FALSE. C 30229000 CONCENTRATE
36300 GO TO 101. C 30230000 CONCENTRATE
36400 END. 3 C 30231000 CONCENTRATE
36500 IF NOT STATION(QUEUED) THEN GO TO 60. C 30232000 CONCENTRATE
36600 END. 2 C 30233000 CONCENTRATE
36700 IF STATION = LASTSTATION THEN C 30234000 CONCENTRATE
36800 IF LINE(QUEUED) THEN C 30235000 CONCENTRATE
36900 LINE(QUEUED) = FALSE. C 30236000 CONCENTRATE
37000 ELSE C 30237000 CONCENTRATE
37100 GO TO 60. C 30238000 CONCENTRATE
37200 GO TO 101. C 30239000 CONCENTRATE
37300 END. 1 C 30240000 CONCENTRATE
37400 STATION=MAXSTATIONS. C 30241000 CONCENTRATE
37500 GO TO 101. C 30242000 CONCENTRATE
37600 %***** C 30243000 CONCENTRATE
37700
37800 C 30244000 CONCENTRATE
37900 REQUEST READRJE: C 30245000 READRJE
38000 C 30246000 READRJE
38100 ERROR[100]= C 30247000 READRJE
38200 TIMEOUT:100, C 30248000 READRJE
38300 LOSSOFCARRIER:100, C 30249000 READRJE
38400 BUFOVFL:106, C 30250000 READRJE
38500 STOPBIT:106, C 30251000 READRJE
38600 PARITY:105, C 30252000 READRJE
38700 ERROR[200]= C 30253000 READRJE
38800 TIMEOUT:303, C 30254000 READRJE
38900 LOSSOFCARRIER:303, C 30255000 READRJE
39000 BUFOVFL:300, C 30256000 READRJE
39100 STOPBIT:300, C 30257000 READRJE
39200 PARITY:300, C 30258000 READRJE
39300 ERROR[300]= C 30259000 READRJE
39400 TIMEOUT:303, C 30260000 READRJE
39500 LOSSOFCARRIER:303, C 30261000 READRJE
39600 BUFOVFL:301, C 30262000 READRJE
39700 STOPBIT:301, C 30263000 READRJE
39800 PARITY:302, C 30264000 READRJE
39900 ERROR[320]= C 30265000 READRJE
40000 TIMEOUT:322, C 30266000 READRJE
40100 LOSSOFCARRIER:322, C 30267000 READRJE
40200 BUFOVFL:322, C 30268000 READRJE
40300 STOPBIT:322, C 30269000 READRJE
40400 PARITY:322, C 30270000 READRJE
40500 ERROR[340]= C 30271000 READRJE
40600 TIMEOUT:343, C 30272000 READRJE
40700 LOSSOFCARRIER:343, C 30273000 READRJE
40800 BUFOVFL:341, C 30274000 READRJE
```

```
40900 STOPBIT:341, C 30275000 READRJE
41000 PARITY:342. C 30276000 READRJE
41100 C 30277000 READRJE
41200 IF CAMEFROMCONTROL THEN C 30278000 READRJE
41300 TERMINATE NORMAL. % PREVENT READ ONCE ONLY C 30279000 READRJE
41400 C 30280000 READRJE
41500 IF GETADDRESS THEN GO TO 1000. C 30281000 READRJE
41600 IF TOG[ESTABLISHED] THEN GO TO 200. C 30282000 READRJE
41700 100: C 30283000 READRJE
41800 INITIATE TRANSMIT. C 30284000 READRJE
41900 IF TRUE THEN C 30285000 READRJE
42000 TRANSMIT ENQ. C 30286000 READRJE
42100 ELSE C 30287000 READRJE
42200 TRANSMIT DLE ENQ. C 30288000 READRJE
42300 FINISH TRANSMIT. C 30289000 READRJE
42400 C 30290000 READRJE
42500 INITIATE RECEIVE. C 30291000 READRJE
42600 RECEIVE[100,ACK:102,ENQ:104,DLE:103]. C 30292000 READRJE
42700 INITIATE TRANSMIT. C 30293000 READRJE
42800 TRANSMIT CHAR. C 30294000 READRJE
42900 FINISH TRANSMIT. C 30295000 READRJE
43000 GO TO 106. C 30296000 READRJE
43100 C 30297000 READRJE
43200 102: C 30298000 READRJE
43300 TOG[ESTABLISHED]=TRUE. C 30299000 READRJE
43400 % INITIALIZE UCSD TYPE CONCENTRATORS C 30300000 READRJE
43500 % C 30301000 READRJE
43600 % C 30302000 READRJE
43700 ESTABLISHTERM = TRUE. C 30303000 READRJE
43800 TERMINATELOGOFF. C 30304000 READRJE
43900 103:% GOT A "DLE":IS THERE AN "ENQ" FOLLOWING? C 30305000 READRJE
44000 RECEIVE(100 MILLI) ENQ[100,FORMATERR:106]. C 30306000 READRJE
44100 C 30307000 READRJE
44200 104: C 30308000 READRJE
44300 INITIATE TRANSMIT. C 30309000 READRJE
44400 TRANSMIT ACK. C 30310000 READRJE
44500 FINISH TRANSMIT. C 30311000 READRJE
44600 GO TO 102. C 30312000 READRJE
44700 C 30313000 READRJE
44800 105: C 30314000 READRJE
44900 IF CHARACTER="4"7F" THEN GO TO 100. C 30315000 READRJE
45000 C 30316000 READRJE
45100 106: C 30317000 READRJE
45200 RECEIVE(100 MILLI)[100]. C 30318000 READRJE
45300 GO TO 106. C 30319000 READRJE
45400 C 30320000 READRJE
45500 200: C 30321000 READRJE
45600 IF TOG[NEEDACK] THEN GO TO 420. C 30322000 READRJE
45700 IF TOG[NEEDNAK] THEN GO TO 430. C 30323000 READRJE
45800 IF TOG[SENTMESSAGE] THEN C 30324000 READRJE
45900 BEGIN 1 C 30325000 READRJE
46000 210: C 30326000 READRJE
46100 INITIATE RECEIVE. C 30327000 READRJE
46200 RECEIVE(12 SEC)[200]. C 30328000 READRJE
46300 END 1 C 30329000 READRJE
46400 ELSE C 30330000 READRJE
46500 BEGIN 1 C 30331000 READRJE
46600 220: C 30332000 READRJE
46700 IF STATION(QUEUED) THEN TERMINATE NOINPUT. C 30333000 READRJE
46800 BADINITIATE = TRUE. % JUST IN CASE... C 30334000 READRJE
46900 LINE(BUSY)=FALSE. C 30335000 READRJE
47000 INITIATE RECEIVE. C 30336000 READRJE
47100 RECEIVE(12 SEC)[200]. C 30337000 READRJE
47200 LINE(BUSY)=TRUE. C 30338000 READRJE
47300 BADINITIATE = FALSE.% ALL CLEAR. C 30339000 READRJE
47400 END. 1 C 30340000 READRJE
47500 221: C 30341000 READRJE
47600 IF CHARACTER="SOH" THEN GO TO 340. C 30342000 READRJE
```

```
47700 IF CHARACTER=LOWERF THEN C 30343000 READRJE
47800 BEGIN 1 C 30344000 READRJE
47900 GOODTRANSNUM=TRUE. C 30345000 READRJE
48000 IF NOT TOG[SENTLOWERF] THEN C 30346000 READRJE
48100 GOODTRANSNUM=FALSE. C 30347000 READRJE
48200 GO TO 320. C 30348000 READRJE
48300 END. 1 C 30349000 READRJE
48400 IF CHARACTER=LOWERU THEN C 30350000 READRJE
48500 BEGIN 1 C 30351000 READRJE
48600 GOODTRANSNUM=TRUE. C 30352000 READRJE
48700 IF TOG[SENTLOWERF] THEN C 30353000 READRJE
48800 GOODTRANSNUM=FALSE. C 30354000 READRJE
48900 GO TO 320. C 30355000 READRJE
49000 END. 1 C 30356000 READRJE
49100 IF CHARACTER=NAK THEN C 30357000 READRJE
49200 BEGIN 1 C 30358000 READRJE
49300 TOG[BEENNAKED]=TRUE. C 30359000 READRJE
49400 GO TO 330. C 30360000 READRJE
49500 END. 1 C 30361000 READRJE
49600 IF CHAR = CAN THEN % C 30362000 READRJE
49700 BEGIN % 1 C 30363000 READRJE
49800 TOG[BEENNAKED] = TRUE, % C 30364000 READRJE
49900 TOG[BEENACKED] = TOG[SENTMESSAGE]. C 30365000 READRJE
50000 END. % 1 C 30366000 READRJE
50100 L[SENT]: C 30367000 READRJE
50200 IF CHAR = SI THEN % RESUME TRANSMISSIONS, C 30368000 READRJE
50300 BEGIN % AFTER GETTING THE 1 C 30369000 READRJE
50400 CNR = SI. % STATION ADDRESS C 30370000 READRJE
50500 GETADDRESS = TRUE. C 30371000 READRJE
50600 TOG[BEENNAKED] = TOG[SENTMESSAGE]. C 30372000 READRJE
50700 IF GOODTRANSNUM THEN C 30373000 READRJE
50800 BEGIN 2 C 30374000 READRJE
50900 TOG[BEENNAKED] = FALSE. C 30375000 READRJE
51000 TOG[BEENACKED] = TOG[SENTMESSAGE]. C 30376000 READRJE
51100 GO 330. C 30377000 READRJE
51200 END. 2 C 30378000 READRJE
51300 GO TO 330. C 30379000 READRJE
51400 END. 1 C 30380000 READRJE
51500 IF CHAR = ENQ THEN C 30381000 READRJE
51600 GO 104. C 30382000 READRJE
51700 IF CHARACTER = DLE THEN C 30383000 READRJE
51800 RECEIVE(100 MILLI)[300,ENQ:104,DLE:315]. C 30384000 READRJE
51900 300: C 30385000 READRJE
52000 LINE(BUSY)=TRUE. C 30386000 READRJE
52100 301: C 30387000 READRJE
52200 RECEIVE(140 MILLI)[300]. C 30388000 READRJE
52300 GO TO 301. C 30389000 READRJE
52400 302: C 30390000 READRJE
52500 IF CHARACTER NE "7F" THEN GO TO 301. C 30391000 READRJE
52600 303: C 30392000 READRJE
52700 LINE(BUSY)=TRUE. C 30393000 READRJE
52800 TOG[BADRESPONSE]=TRUE. C 30394000 READRJE
52900 TERMINATE NOINPUT. C 30395000 READRJE
53000 310: C 30396000 READRJE
53100 RECEIVE(100 MILLI)[300,DLE:315]. C 30397000 READRJE
53200 GO TO 301. C 30398000 READRJE
53300 315: C 30399000 READRJE
53400 RECEIVE(100 MILLI)[300,ENQ:104]. C 30400000 READRJE
53500 GO TO 301. C 30401000 READRJE
53600 320: C 30402000 READRJE
53700 RECEIVE(100 MILLI)[320,ACK:321,SO:326,SUB:327] C 30403000 READRJE
53800 322: C 30404000 READRJE
53900 TOG[BADRESPONSE]=TRUE. C 30405000 READRJE
54000 GO TO 301. C 30406000 READRJE
54100 327: % SUB, ACKED BUT SUSPEND TRANSMISSIONS C 30407000 READRJE
54200 WAITING = TOG[SENTMESSAGE]. C 30408000 READRJE
54300 321: C 30409000 READRJE
54400 TOG[BEENNAKED]=TRUE. C 30410000 READRJE
```



```

54500      IF GOODTRANSNUM THEN                C 30411000 READRJE
54600      BEGIN                               1 C 30412000 READRJE
54700      TOG[BEENACKED]=TOG[SENTMESSAGE].    C 30413000 READRJE
54800      TOG[BEENNAKED]=FALSE.              C 30414000 READRJE
54900      END.                                1 C 30415000 READRJE
55000      326: % SO NAKED TRANSMISSION        C 30416000 READRJE
55100      IF CHAR = SO THEN % AND SUSPEND TRANS C 30417000 READRJE
55200      BEGIN                               1 C 30418000 READRJE
55300      TOG[BEENNAKED] = TRUE.             C 30419000 READRJE
55400      WAITING = TRUE.                    C 30420000 READRJE
55500      END.                                1 C 30421000 READRJE
55600      330:                                C 30422000 READRJE
55700      TERMINATE NOINPUT.                  C 30423000 READRJE
55800      340:                                C 30424000 READRJE
55900      INITIALIZE BCC.                     C 30425000 READRJE
56000      GOODTRANSNUM=FALSE.                 C 30426000 READRJE
56100      RECEIVE(100 MILLI)[340].           C 30427000 READRJE
56200      IF TOG[SENTLOWERF] THEN           C 30428000 READRJE
56300      BEGIN                               1 C 30429000 READRJE
56400      IF CHAR=LOWERF THEN                 C 30430000 READRJE
56500      GOODTRANSNUM=TRUE.                 C 30431000 READRJE
56600      END ELSE                           1 C 30432000 READRJE
56700      BEGIN                               1 C 30433000 READRJE
56800      IF CHAR=LOWERU THEN                 C 30434000 READRJE
56900      GOODTRANSNUM=TRUE.                 C 30435000 READRJE
57000      END.                                1 C 30436000 READRJE
57100      RECEIVE(100 MILLI)[340].           C 30437000 READRJE
57200      NEWTRANSNUM=FALSE.                 C 30438000 READRJE
57300      IF CNR NE CHARACTER THEN           C 30439000 READRJE
57400      BEGIN                               1 C 30440000 READRJE
57500      CNR=CHARACTER.                      C 30441000 READRJE
57600      NEWTRANSNUM=TRUE.                  C 30442000 READRJE
57700      END.                                1 C 30443000 READRJE
57800      RECEIVE(100 MILLI)STX[340,FORMATERR:341]. C 30444000 READRJE
57900      TALLY[0]=BCC.                      C 30445000 READRJE
58000      GETADDRESS=TRUE.                   C 30446000 READRJE
58100      TERMINATE NOINPUT.                 C 30447000 READRJE
58200      1000:                                C 30448000 READRJE
58300      GETADDRESS=FALSE.                  C 30449000 READRJE
58400      IF TOG[NEEDNAK] THEN GO TO 341.     C 30450000 READRJE
58500      BCC=TALLY[0].                      C 30451000 READRJE
58600      IF CNR = SI THEN %EAT UNWANTED      C 30452000 READRJE
58700      BEGIN                               1 C 30453000 READRJE
58800      CNR = 0. % RESUMES                   C 30454000 READRJE
58900      IF WAITING THEN % DEFAULT           C 30455000 READRJE
59000      BEGIN                               2 C 30456000 READRJE
59100      WAITING = FALSE.                    C 30457000 READRJE
59200      FREQCNTR = 0.                       C 30458000 READRJE
59300      END.                                2 C 30459000 READRJE
59400      RECEIVE(100 MILLI)[TIMEOUT:LNOCNR,LOSSOFFCARRIER: C 30460000 READRJE
59500      ,BUFOVFL:343,STOPBIT:341,PARITY:342]. C 30461000 READRJE
59600      IF CHAR NEQ 4"FF" THEN % PAD        C 30462000 READRJE
59700      BEGIN                               2 C 30463000 READRJE
59800      CNR = CHAR.                          C 30464000 READRJE
59900      NEWTRANSNUM = TRUE.                 C 30465000 READRJE
60000      TOG[NEEDACK] = TRUE.                C 30466000 READRJE
60100      END.                                2 C 30467000 READRJE
60200      LNOCNR:                             C 30468000 READRJE
60300      GO 330.                             C 30469000 READRJE
60400      END.                                1 C 30470000 READRJE
60500      1100:                                C 30471000 READRJE
60600      GETSPACE[341].                      C 30472000 READRJE
60700      RECEIVE(100 MILLI)TEXT[340,        C 30473000 READRJE
60800      CONTROL:NULL,                       C 30474000 READRJE
60900      ENDOFBUFFER:345,                   C 30475000 READRJE
61000      BACKSPACE:NULL,                   C 30476000 READRJE
61100      ETX].                              C 30477000 READRJE
61200      RECEIVE(100 MILLI)BCC[340,BCCERR:341]. C 30478000 READRJE

```

```
61300
61400
61500
61600
61700
61800
61900
62000
62100
62200
62300
62400
62500
62600
62700
62800
62900
63000
63100
63200
63300
63400
63500
63600
63700
63800
63900
64000
64100
64200
64300
64400
64500
64600
64700
64800
64900
65000
65100
65200
65300
65400
65500
65600
65700
65800
65900
66000
66100
66200
66300
66400
66500
66600
66700
66800
66900
67000
67100
67200
67300
67400
67500
67600
67700
67800
67900
68000

TOG[NEEDACK]=TRUE.
TOG[BEENNAKED]=TRUE.
IF GOODTRANSM THEN
  BEGIN
    TOG[BEENNAKED]=TOG[SENTMESSAGE].
    TOG[BEENNAKED]=FALSE.
  END.
WAITING = FALSE.
FREQNTR = 0.
IF NEWTRANSM THEN IF
  NOT BADADDRESS THEN
  TERMINATE NORMAL.
  TERMINATE NOINPUT.
345:
  TALLY[0] = BCC.
  FREQNTR = CHAR.
  TERMINATE BLOCK.
  BCC = TALLY[0].
  CHAR = FREQNTR.
  GETSPACE[341]. %REV 9/24/77 2.9C
  STORE.
  GO 1100.
341:
  RECEIVE(100 MILLI)[340].
  GO TO 341.
342:
  IF CHARACTER NE 4"7F" THEN GO TO 341.
343:
  IF NEWTRANSM THEN CNR=0.
  TOG[NEEDACK] = FALSE. % MAY HAVE BEEN
  IF TOG[SENTMESSAGE] THEN IF % IN
  GOODTRANSM THEN % CONTROL
  GO TO 430.
  IF NOSPACE THEN
  GO NOSPACE.
  IF WAITING THEN % RESET ON TIMEOUT
  WAITING = FALSE ELSE
  TOG[NEEDNAK]=TRUE.
  TERMINATE NOINPUT.
420:
  TOG[NEEDACK]=FALSE.
  INITIATE TRANSMIT.
  CHARACTER=CNR.
  TRANSMIT.
  TRANSMIT ACK.
  FINISH TRANSMIT.
  GO TO 220.
430:
  IF NOSPACE THEN
  BEGIN
    NOSPACE:
    INITIATE TRANSMIT.
    TRANSMIT SOH.
    INITIALIZE BCC.
    TRANSMIT NUL.
    CHAR = LOWERU.
    IF TOG[SENTLOWERF] THEN
    CHAR = LOWERF.
    TRANSMIT.
    TRANSMIT STX "00".
    IF UCSDTERMINAL THEN
    TRANSMIT "0".
    TRANSMIT RJEWAITMSG.
```

```
C 30479000 READRJE
C 30480000 READRJE
C 30481000 READRJE
C 30482000 READRJE
1 C 30483000 READRJE
C 30484000 READRJE
C 30485000 READRJE
1 C 30486000 READRJE
C 30487000 READRJE
C 30488000 READRJE
C 30489000 READRJE
C 30490000 READRJE
C 30491000 READRJE
C 30492000 READRJE
C 30493000 READRJE
C 30494000 READRJE
C 30495000 READRJE
C 30496000 READRJE
C 30497000 READRJE
C 30498000 READRJE
C 30499000 READRJE
C 30500000 READRJE
C 30501000 READRJE
C 30502000 READRJE
C 30503000 READRJE
C 30504000 READRJE
C 30505000 READRJE
C 30506000 READRJE
C 30507000 READRJE
C 30508000 READRJE
C 30509000 READRJE
C 30510000 READRJE
C 30511000 READRJE
C 30512000 READRJE
C 30513000 READRJE
C 30514000 READRJE
C 30515000 READRJE
C 30516000 READRJE
C 30517000 READRJE
C 30518000 READRJE
C 30519000 READRJE
C 30520000 READRJE
C 30521000 READRJE
C 30522000 READRJE
C 30523000 READRJE
C 30524000 READRJE
C 30525000 READRJE
C 30526000 READRJE
C 30527000 READRJE
C 30528000 READRJE
C 30529000 READRJE
C 30530000 READRJE
C 30531000 READRJE
C 30532000 READRJE
1 C 30533000 READRJE
C 30534000 READRJE
C 30535000 READRJE
C 30536000 READRJE
C 30537000 READRJE
C 30538000 READRJE
C 30539000 READRJE
C 30540000 READRJE
C 30541000 READRJE
C 30542000 READRJE
C 30543000 READRJE
C 30544000 READRJE
C 30545000 READRJE
C 30546000 READRJE
```



```

74900          TOGS=0.          %REINITIALIZE          C 30617000 WRITERJE
75000          TALLY[2]=0.      C 30618000 WRITERJE
75100          TERMINATE NORMAL. C 30619000 WRITERJE
75200          END.              1 C 30620000 WRITERJE
75300          TRANSMIT SOH.     C 30621000 WRITERJE
75400          INITIALIZE BCC.   C 30622000 WRITERJE
75500          CHARACTER=CNR.    C 30623000 WRITERJE
75600          TRANSMIT.         C 30624000 WRITERJE
75700          CHARACTER=LOWERU. C 30625000 WRITERJE
75800          IF TOG[SENTLOWERF] THEN C 30626000 WRITERJE
75900          CHARACTER=LOWERF. C 30627000 WRITERJE
76000          TRANSMIT.         C 30628000 WRITERJE
76100          TRANSMIT STX.     C 30629000 WRITERJE
76200          TRANSADDRESS(LINE(TALLY[0])). 1 C 30630000 WRITERJE
76300          IF UCS[TERMINAL] THEN C 30631000 WRITERJE
76400          CARRIAGECONTROL.  1 C 30632000 WRITERJE
76500          RJEDELETEBLANK(LINE(TALLY[0]),STATION(TALLY)).1 C 30633000 WRITERJE
76600          C 30634000 WRITERJE
76700          TRANSMIT ETX.     C 30635000 WRITERJE
76800          TRANSMIT BCC.     C 30636000 WRITERJE
76900          FINISH TRANSMIT.  C 30637000 WRITERJE
77000          C 30638000 WRITERJE
77100          LASTSTATION=255.  C 30639000 WRITERJE
77200          TOG[NEEDACK]=FALSE. C 30640000 WRITERJE
77300          TOG[NEEDNAK]=FALSE. C 30641000 WRITERJE
77400          TOG[SENTMESSAGE]=TRUE. C 30642000 WRITERJE
77500          TOG[BEENACKED]=FALSE. C 30643000 WRITERJE
77600          TOG[BEENNAKED]=FALSE. C 30644000 WRITERJE
77700          FREQCNTR = STATION(FREQUENCY). C 30646000 WRITERJE
77800          TERMINATE.        C 30647000 WRITERJE
77900          %*****          C 30648000 WRITERJE
78000
78100          MODEM SYSCOMC:      C 40000000 WRITERJE
78200          ADAPTER = 4,6,7,8,9,11,12,13,14,15. C 40001000
78300          NOISEDELAY = 0.    C 40002000
78400          TRANSMITDELAY = 125 MILLI. C 40003000
78500          RECEIVEDDELAY = 10 MILLI.  C 40004000
78600          TERMINAL DEFAULT RJE:    C 50000000
78700          CONTROL = CONCENTRATE.   C 50001000
78800          SCREEN = TRUE.            C 50002000
78900          ADDRESS = 2.             C 50003000
79000          WIDTH = 420.            C 50004000
79100          BUFFER = 420.           C 50005000
79200          TIMEOUT = 1500 MILLI.    C 50006000
79300          % TRANSMISSION = 1.      C 50007000
79400          TURNAROUND= 50 MILLI.    C 50008000
79500          ADAPTER = 4,6,7,8,9,11,12,13,14,15,20. C 50009000
79600          CODE = ASC67.           C 50010000
79700          BLOCK = FALSE.          C 50011000
79800          DUPLEX = FALSE.          C 50012000
79900          ICTDELAY = 0.           C 50013000
80000          PAGE = NULL.           C 50014000
80100          TERMINAL RJEAMOD:      C 50015000
80200          ADAPTER = 20,23,26,9,12,15,14. C 50016000
80300          DEFAULT = RJE.          C 50017000
80400          REQUEST = READRJE:RECEIVE, C 50018000
80500          WRITERJE:TRANSMIT.      C 50019000
80600          PARITY = VERTICAL:ODD,   C 50020000
80700          HORIZONTAL:EVEN.        C 50021000
80800          BACKSPACE = 4"FF".      C 50022000
80900          TURNAROUND = 140 MILLI. C 50023000
81000          SYNCS = TRUE.           C 50024000
81100          TERMINAL RJEPTAMOD:     C 50025000
81200          ADAPTER =7,23,8,9,12,14,15,20,24. C 50026000
81300          DEFAULT = RJE.          C 50027000
81400          REQUEST = READRJE:RECEIVE, C 50028000
81500          WRITERJE:TRANSMIT.      C 50029000
81600          PARITY = VERTICAL:ODD,   C 50030000

```

81700		HORIZONTAL: EVEN.	C 50031000
81800	BACKSPACE	= 4"FF".	C 50032000
81900	TURNAROUND	= 140 MILLI.	C 50033000
82000	SYNCS	= TRUE.	C 50034000
82100	TERMINAL RJE: BMOD:		C 50034500
82200	ADAPTER	= 20, 23, 26, 9, 12, 15, 14.	C 50035000
82300	DEFAULT	= RJE.	C 50036000
82400	REQUEST	= READRJE: RECEIVE,	C 50037000
82500		WRITERJE: TRANSMIT.	C 50038000
82600	PARITY	= VERTICAL: ODD,	C 50039000
82700		HORIZONTAL: EVEN.	C 50040000
82800	BACKSPACE	= 4"FF".	C 50041000
82900	TURNAROUND	= 560 MILLI.	C 50042000
83000	SYNCS	= TRUE.	C 50043000
83100	TERMINAL RJE: PTBMOD:		C 50044000
83200	ADAPTER	= 7, 23, 8, 9, 12, 14, 15, 20, 26.	C 50045000
83300	DEFAULT	= RJE.	C 50046000
83400	REQUEST	= READRJE: RECEIVE,	C 50047000
83500	WRITERJE: TRANSMIT.		C 50048000
83600	PARITY	= VERTICAL: ODD,	C 50049000
83700		HORIZONTAL: EVEN.	C 50050000
83800	BACKSPACE	= 4"FF".	C 50051000
83900	TURNAROUND	= 560 MILLI.	C 50052000
84000	SYNCS	= TRUE.	C 50053000
84100			
84200	STATION DEFAULT RJE: STADEF:		C 60000000
84300	ENABLE INPUT	= TRUE.	C 60001000
84400	LOGIN	= TRUE.	C 60002000
84500	LOGICAL ACK	= FALSE.	C 60003000
84600	WRAP AROUND	= TRUE.	C 60004000
84700	MY USE	= INPUT, OUTPUT.	C 60005000
84800	MCS	= SYSTEM/B771.	C 60006000
84900	RETRY	= 10.	C 60007000
85000	CONTROL	= QM.	C 60008000
85100	STATION DEFAULT ARJEM1:		C 60009000
85200	DEFAULT	= RJE: STADEF.	C 60010000
85300	TERMINAL	= RJE: AMOD.	C 60011000
85400	STATION DEFAULT ARJ1:		C 60012000
85500	DEFAULT	= RJE: STADEF.	C 60013000
85600	TERMINAL	= RJE: PTAMOD.	C 60014000
85700	STATION DEFAULT ARJEM2:		C 60014500
85800	DEFAULT	= RJE: STADEF.	C 60015000
85900	TERMINAL	= RJE: BMOD.	C 60016000
86000	STATION DEFAULT ARJ2:		C 60017000
86100	DEFAULT	= RJE: STADEF.	C 60018000
86200	TERMINAL	= RJE: PTBMOD.	C 60019000
86300			
86400	(LSN = 00002)	STATION AUTOSERVICIO/CSC:	C 60021000
86500	DEFAULT	= ARJEM1.	C 60022000
86600	ADAPTER	= 15.	C 60023000
86700	MODEM	= SYSCOMC.	C 60024000
86800	ADDRESS	= "00".	C 60025000
86900	(LSN = 00003)	STATION AUTOSERVICIO/CSCSD:	C 60026000
87000	DEFAULT	= ARJEM1.	C 60027000
87100	ADAPTER	= 15.	C 60028000
87200	MODEM	= SYSCOMC.	C 60029000
87300	ADDRESS	= "01".	C 60030000
87400	WIDTH	= 72.	C 60031000
87500	CONTROL	= QM.	C 60032000
87600	(LSN = 00004)	STATION AUTOSERVICIO/CSCCR:	C 60033000
87700	DEFAULT	= ARJEM1.	C 60034000
87800	ADAPTER	= 15.	C 60035000
87900	MODEM	= SYSCOMC.	C 60036000
88000	ADDRESS	= "02".	C 60037000
88100	(LSN = 00005)	STATION AUTOSERVICIO/CSCPLP:	C 60038000
88200	DEFAULT	= ARJ1.	C 60039000
88300	ADAPTER	= 15.	C 60040000
88400	MODEM	= SYSCOMC.	C 60041000

```
88500 ADDRESS = "03". C 60042000
88600
88700 LINE LX11: C 70000000
88800 ADAPTER = 5(MODEM). C 70001000
88900 ADDRESS = 1:1:1. C 70002000
89000 STATION = AUTOSERVICIO/CSC, C 70003000
89100 AUTOSERVICIO/CSCSC, C 70004000
89200 AUTOSERVICIO/CSCCR, C 70005000
89300 AUTOSERVICIO/CSCPL. C 70006000
89400 MODEM = SYSCOMC. C 70007000
89500 DCP 0: C 80000000
89600 EXCHANGE = 1. C 80001000
89700 MEMORY = 40%. C 80002000
89800 TERMINAL = RJEAMOD(MSGSPACE=20), C 80003000
89900 RJEPTAMOD(MSGSPACE=20), C 80004000
90000 RJEEMOD(MSGSPACE=20), C 80005000
90100 RJEPTBMOD(MSGSPACE=20), C 80006000
90200 OPTIONS = LOCALTABLES. C 80007000
90300 DCP 1: C 80008000
90400 EXCHANGE = 0. C 80009000
90500 MEMORY = 40%. C 80010000
90600 TERMINAL = RJEAMOD(MSGSPACE=20), C 80011000
90700 RJEPTAMOD(MSGSPACE=20), C 80012000
90800 RJEEMOD(MSGSPACE=20), C 80013000
90900 RJEPTBMOD(MSGSPACE=20), C 80014000
91000 OPTIONS = LOCALTABLES. C 80015000
91100
```

```
91200 =====
91300 NO SYNTAX ERRORS DETECTED; NO WARNINGS ISSUED;
91400 PROGRAM SIZE = 900 CARDS, 4535 SYNTACTIC ITEMS.
91450 125 DISK SEGMENTS FOR NIF
91500 DATACOM FILE PREFIX: (CSCF)CANDE/COE771 ON CSCS
91600 =====
```

```
91700 ENABLE INPUT SPACE MEMORY REQUIREMENTS:
91800 INDEX AREA INITIAL
91900 OF SIZE NUMBER OF
92000 AREA (WORDS) AREAS
92100
92200 1 80 160
92300 2 80 1
92400 TOTAL 12880 161
92500 =====
```

```
92600 SUMMARY FOR DCP 0:
92700 NO ERROR WAS DISCOVERED.
92800 LOCAL MEMORY REQUIRES 1720 WORDS AS FOLLOWS:
92900 DCP CODE 1651
93000 CONTROL BLOCK 72 BASE ADDRESS = 0045 (HEX)
93100 ADDRESS BLOCK 4
93200 STRING TABLE 3
93300 LOCAL TABLES 40
93400 UNUSED 0
93500 DCP CODE REQUIRES 34 DISK SEGMENTS
93600 BRANCH TABLE SIZE = 40 WORDS
93700 =====
```

```
93800 SUMMARY FOR DCP 1:
93900 NO ERROR WAS DISCOVERED.
94000 LOCAL MEMORY REQUIRES 1720 WORDS AS FOLLOWS:
94100 DCP CODE 1657
94200 CONTROL BLOCK 4 BASE ADDRESS = 0045 (HEX)
94300 ADDRESS BLOCK 4
94400 STRING TABLE 3
94500 LOCAL TABLES 40
94600 UNUSED 0
94700 DCP CODE REQUIRES 34 DISK SEGMENTS
94800 BRANCH TABLE SIZE = 40 WORDS
94900 =====
```

```
95000 ELAPSED TIME = 45.10 SECONDS, PROCESSOR TIME = 14.18 SECONDS,
95100 I/O TIME = 12.16 SECONDS.
```

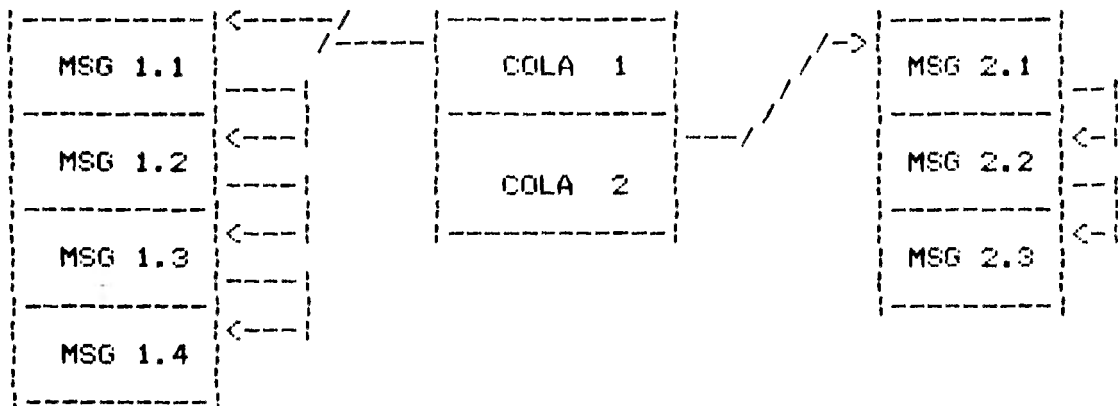
95200

=====

2.3.- Funcionamiento de la comunicación de datos en la B6700.

Los vehículos más importantes en la transferencia de información, entre los elementos que forman el sistema de comunicación de datos de la B6700, son los mensajes.

Los mensajes son áreas de memoria que se encuentran ligadas entre sí por apuntadores y forman en conjunto las colas.



Estructura general de las colas de mensajes en la B6700

Existen varias colas en el sistema. Cada DCP tiene un par de ellas que son la "Request Queue" y la "Result Queue".

En la "Request Queue" son puestos los mensajes que se mandan hacia los DCP's. En la "Result Queue" se obtienen los mensajes que llegan del DCP hacia el sistema central.

Cada estación tiene una cola que se llama "Station Queue", en donde se ponen todos los mensajes que van a esa

estación. Cada MCS tiene una cola llamada "Primary Queue" y ninguna, o varias colas, llamadas "Current Queue".

Estas colas sirven para que el sistema de comunicación de datos se comuniquen con el MCS. De esta forma los mensajes forman la información que es transferida entre programas de aplicación y el MCS, programas de aplicación y el DCC, el DCC y el MCS, etc,etc.

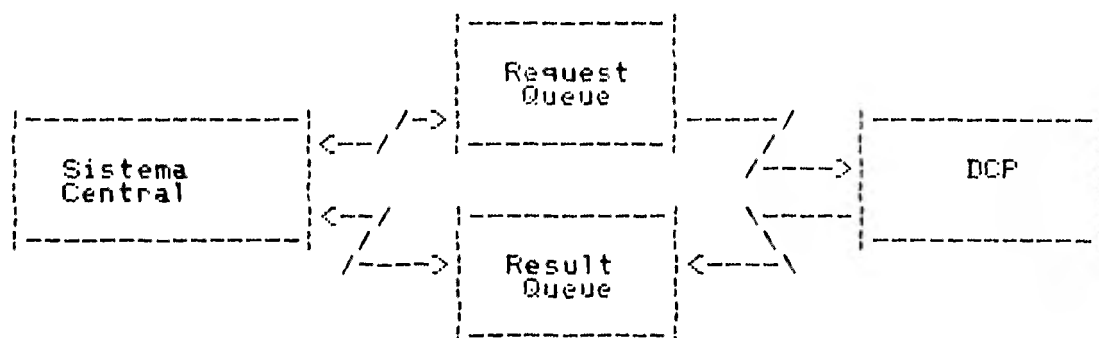


Diagrama de la "Result Queue" y la "Request Queue".

La transferencia de información no involucra necesariamente movimientos físicos, sino que al pasar un mensaje de un lugar a otro, solamente las cadenas o apuntadores de éste son cambiados.

Todos los mensajes constan de un número entero de palabras, de 6 caracteres y contienen un encabezado que en general es de seis (6) palabras, a continuación del encabezado está una parte de texto, que es el mensaje en sí.

Todas las transferencias de información en el sistema se hacen a través de áreas de memoria, que en el caso de un programa en una escritura le pasa al MCP el dato en una área llamada "Program Buffer", lo mismo sucede al dar un "Read", el área es puesta en el archivo correspondiente por las llamadas de "I/O". Entonces el registro es puesto en el Program Buffer y el I/O intrínseco copia a éste dentro de la parte de texto de un mensaje y lo encadena a la cola apropiada. Si este es puesto en la "Request Queue" el "Buffer" permanece ocupado hasta que el DCP lo saca de la "Request Queue" para ponerlo en una terminal. En caso de que el MCS que controla la estación decida participar en las operaciones de I/O, este mensaje es pasado a una cola del MCS para que éste lo ponga en la estación correspondiente y pueda

hacer trabajos de edición o toma de decisiones.

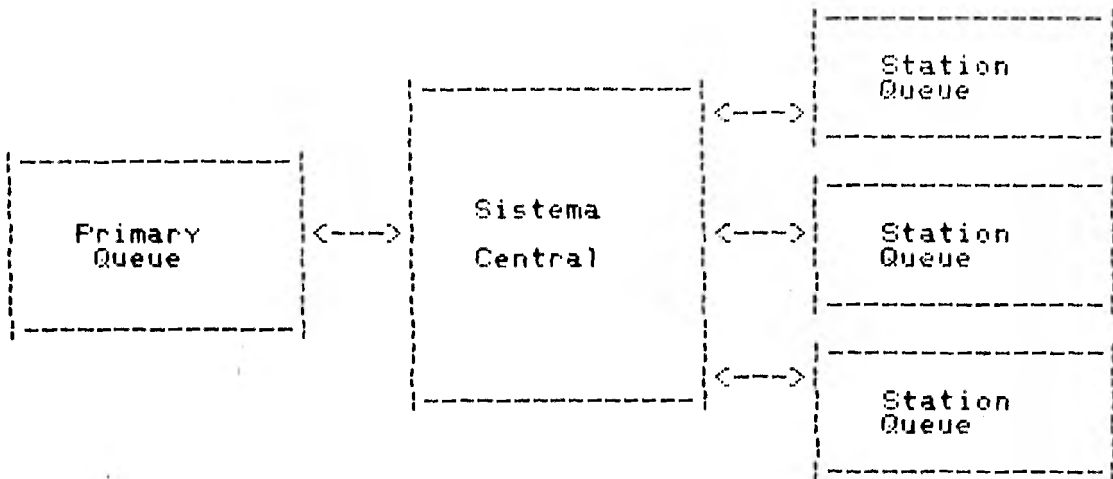


Diagrama de la "Primary Queue" y las "Stations Queues".

El MCS, el DCC y los I/O intrínsecos se comunican entre sí a través de las colas.

Las denominadas "DCALGOL Queue" y las colas de archivos remotos son de una misma estructura, mientras que las "Result Queue", "Request Queue" y "Station Queue", son de otra estructura.

Las colas de archivos remotos y las de DCALGOL tienen un único descriptor de datos en el "Stack". Este descriptor de datos apunta a un conjunto de palabras contiguas llamadas "Hidden Message", este mensaje contiene estadísticas sobre la cola como: número de mensajes en la cola y el número de palabras en memoria de todos los mensajes que contiene.

Cada DCP tiene su "Request Queue", de la cual extrae los mensajes y estas colas son el único medio por el cual los mensajes son transmitidos del sistema central a los DCP's y viceversa.

Cada mensaje debe ser enrutado a uno de los varios MCS's de la "Result Queue" siempre y cuando éste pertenezca a una estación bajo su control. Todos estos ruteos de mensajes

los hace el DCC, que para simplificar sus funciones tiene un proceso asíncrono que sirve a cada DCP con un par de "Request Queue" para cada uno. La función más importante de este proceso es extraer información de la "Request Queue".

Algunas veces también inserta información en la "Request Queue", siendo ésto más comunmente hecho por los "I/O Intrinsic" del MCP y los "DCWrite Intrinsic" usados por el MCS para insertar mensajes en la "Request Queue".

La cola en la cual recibe un MCS información o comunicación del DCC es la "Primary Queue" y cada MCS al comenzar a funcionar tiene que iniciar esta "Primary Queue" para que el sistema tenga el medio de comunicarse con el MCS correspondiente. El MCS se comunicará a través de los "DCWrite Intrinsic", que son llamadas instrucciones de DCALGOL para ejecutar diferentes funciones de acuerdo a la información que se pase a estos intrínsecos.

Cuando se tecléa un carácter desde una terminal remota del sistema, tienen lugar los siguientes eventos:

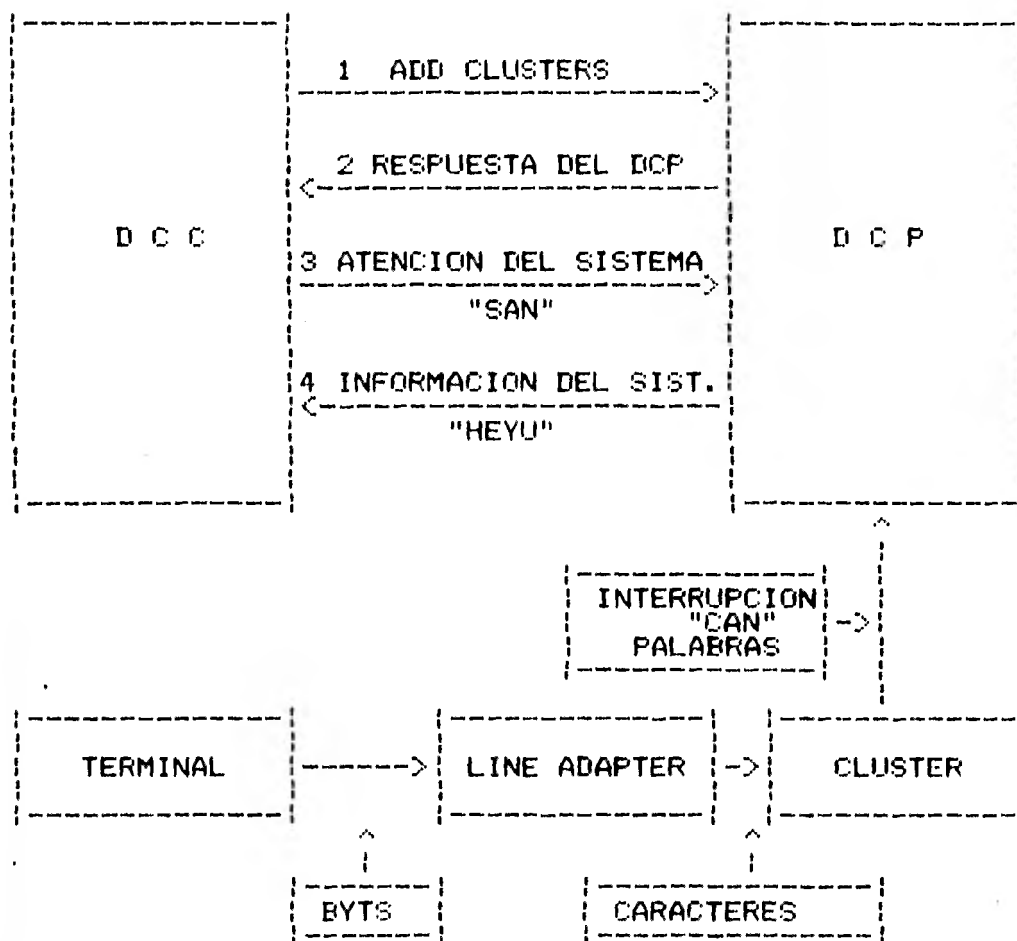
- 1.- El teclado de la terminal remota envía una señal codificada al microprocesador de la terminal, indicándole que fué tecléado un carácter y el código ASCII de dicho carácter, el CPU de la terminal (si lo tiene la terminal), envía este código en forma paralela al puerto encargado de transmitir este carácter en forma serial, y además convertirlo a niveles de voltaje lógicos de la interface RS-232C (ver apéndice IV).
- 2.- La señal serie se propaga por los conductores que forman un canal de comunicación entre la terminal y la computadora central.

- 3.- El line-adapter detecta los niveles lógicos de tensión que se propagaron y llegaron a él en forma de señales RS-232C. Estos niveles de tensión son convertidos en bits.
- 4.- El adapter-cluster recibe del line-adapter los bits que forman un carácter, y los acumula hasta que puede formar un carácter.
- 5.- El DCP recibe los caracteres que le pasa el adapter-cluster, hasta acumular una palabra de 6 (seis) caracteres, a los cuales se les denomina mensaje.
- 6.- Cuando se encuentra formado el mensaje, el software que se encarga de la supervisión y control de la red, previamente definido en NDL y que se encuentra corriendo en el DCP, "reconoce" el mensaje e investiga a que MCS está adscrita la terminal de donde proviene el mensaje, toda esta información es proporcionada al DCC.
- 7.- El DCC checa la validez del mensaje y de las instrucciones que le pasa el DCP a el MCS respectivo. Si el mensaje es válido, lo pone en la "request queue" del MCS respectivo.
- 8.- El MCS es el encargado de "reconocer" e interpretar la información proporcionada en el mensaje depositado en su "request queue".
- 9.- Si se trata de algún comando para que el MCS lo realice, (tal como la asignación de un archivo

en disco, la compilación de algún programa, etc.) ejecuta dicha instrucción si está capacitado para ella, si no, informará al usuario que se encuentra en la terminal remota, que no está capacitado para efectuar dicha operación.

- 10.- Si se trata de un mensaje para algún programa que esté corriendo a través de este MCS, el mensaje es pasado al programa respectivo.

Para el caso en que el usuario que está operando la terminal remota, ejecute un programa que es interactivo, el proceso de comunicación de el programa con el usuario, será inverso al descrito en los incisos anteriores.



Proceso de comunicación de datos en la B6700.

* referencias 2.1 y 2.2

2.4.- CANDE

CANDE es un MCS proporcionado por BURROUGHS, está escrito en DCALGOL; el propósito específico de este MCS es el comunicar a un usuario situado en una terminal remota, con la computadora central.

Se disponen de comandos por medio de los cuales es posible:

- Edición y corrección de archivos.
- Ejecución de programas.
- Lanzado de trabajos.
- Preguntar el estado de una cierta línea.
- Preguntar los recursos de máquina que un programa ha consumido hasta un determinado momento.
- Reconfigurar la red dinámicamente.
- Investigar los recursos de disco consumidos.
- Comunicarse con otro usuario conectado a la red de teleproceso.
- Descontinuar trabajos lanzados por el usuario.

Para una información más compleja y detallada consultar la referencia mencionada abajo.

* referencia 2.8

2.5.- Flujo de trabajos WFL (WORK FLOW LANGUAGE).

El lenguaje de flujo de trabajos (WFL), es el medio por el cual es posible indicar y describir al sistema BURROUGHS, cuales trabajos (JOBS) deberá de ejecutar y en qué secuencia.

Este lenguaje permite al usuario describir cada trabajo como un conjunto de tareas relacionadas entre sí, y que serán ejecutadas. Se puede definir condiciones en las que, si una condición se cumple, se ejecuta otra condición. Es posible definir además la ejecución serie y/o paralelo de los trabajos especificados.

WFL es un compilador del tipo real; el cual produce código para controlar las tareas que el usuario describe en los "JOBS". Por lo tanto los multicitados "JOBS" son programas escritos por el usuario en WFL. El compilador de WFL puede realizar las siguientes funciones:

- a).- Corrija sintácticamente los enunciados de un programa en WFL.
- b).- Genera código para manejar las tareas que constituyen un "JOB" que se especifican en él.
- c).- Genera el archivo en disco necesario para la ejecución del "JOB".

* referencia 2.6

2.6.- Características de las líneas.

Para la conexión de dos dispositivos de comunicación, no importa el tipo de que se trate, es necesario contar con un medio físico que las una. Este puede ser la atmósfera terrestre, una línea telefónica, línea directa, fibra óptica, o cualquier otro medio; pero por ser el segundo el más difundido en la actualidad, partiremos de que éste será el medio para comunicar a las dos computadoras bajo estudio.

Una línea telefónica es una estructura conductora formada por pares conductores, y capaz de transmitir energía eléctrica entre dos puntos.

Podemos modelar a una línea de transmisión en general de la siguiente manera:

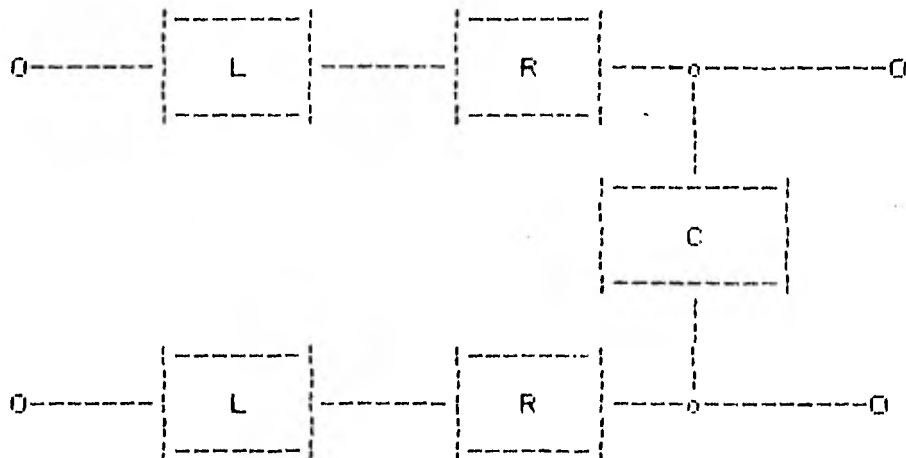


Diagrama de Bloques de una sección de línea de transmisión.

Existen características básicas de una línea telefónica y son: resistencia, capacitancia, inductancia y reactancia, siendo esta una combinación de las dos primeras.

2.6.1.- Resistencia.

Es la oposición al flujo de electrones a través del circuito, todo conductor, aislante o cualquier otro tipo de material, presenta resistencia. Una tensión o fuerza electromotriz es necesaria para vencer la resistencia y forzar a la corriente a fluir a través de esta resistencia. Cuando esto sucede se produce calor. A la cantidad de calor disipado corresponde una potencia disipada y se expresa en watts.

En líneas de transmisión largas, la resistencia de los dos conductores es una parte considerable del total de la resistencia de todo el circuito.

La resistencia de un conductor depende de diversos factores. Uno de ellos es el material o metal que se utiliza para construir el conductor.

La siguiente tabla muestra la resistencia de algunos metales comparados con el cobre.

Conductor	Resistencia Relativa
Plata	0.92
Oro	1.38
Aluminio	1.59
Acero	8.62

El cobre es el material comunmente más utilizado en la fabricación de conductores, por lo tanto se utiliza para comparación, y tiene una resistencia relativa de uno (1).

La longitud y diámetro de los conductores afectan de la misma manera la pérdida de potencia. Si la sección transversal de un conductor se incrementa, la resistencia disminuye. Por otra parte, si el diámetro se mantiene constante y la longitud se incrementa, la resistencia total crecerá proporcionalmente. Por ejemplo, si un cable de una milla de largo tiene una resistencia de 3 ohms, 5 millas del mismo cable tendrán una resistencia de 15 ohms.

Si la frecuencia de las señales alternas aplicadas a un conductor se incrementa, la corriente tiende a fluir muy cerca de la superficie, lejos del centro físico del conductor. Por lo tanto, no toda el area seccional es utilizada, la resistencia de un conductor se incrementa proporcionalmente con la frecuencia. Este fenómeno es conocido como "efecto de piel" (Skin effect), y es muy importante en sistemas de transmisión.

2.6.2.- Reactancia.

Una segunda propiedad de las líneas de transmisión, que es también muy importante en los circuitos de comunicación, es la Reactancia. Está simbolizada por la letra "X", y se mide como la oposición al flujo de corrientes alternas. La cantidad de reactancia es medida en Ohms. Son dos los tipos de reactancias que existen: la reactancia asociada a los efectos inductivos (reactancia inductiva X_L) y la asociada a los efectos capacitivos (reactancia capacitiva X_C).

2.6.3.- Inductancia.

Todos los conductores, no importa su longitud, tienen inductancia. Se considera que está en serie con el conductor, y se incrementa proporcionalmente a la longitud del conductor. Después de una cierta longitud, las líneas de transmisión pueden ser divididas en segmentos de línea o simplemente secciones; cada una compuesta de pequeñas inductancias, e idénticas a las demás inductancias de otras secciones.

2.6.4.- Capacitancia.

Todos los conductores, presentan un cierto grado de capacitancia entre ellos. Esta capacitancia es producida por el paralelismo que existe entre los conductores a lo largo de cierta distancia; esta distancia puede ser desde unos centímetros hasta cientos de kilómetros. Esta capacitancia produce reactancia capacitiva.

2.6.5.- Impedancia.

La impedancia es otra característica importante de las líneas de transmisión. Esta se representa por la letra "Z" y se expresa en Ohms. Esta es una combinación de la resistencia y la reactancia.

Las características antes mencionadas de las líneas de transmisión ocasionan en la señal transmitida: desfazamiento, y atenuación.

* referencias 2.3 y 2.4

2.7.- Modems y otros dispositivos auxiliares.

En la sección anterior vimos como es afectada la calidad de transmisión, con una línea de transmisión de cierta longitud. Se introduce ruido, defasamiento, retardos, atenuación, etc. Estos problemas se presentan con mayor fuerza a medida que se incrementa la longitud de las líneas. Para poder usar estas líneas se utilizan los MODEMS. MODEM es una contracción de MODulador DEModulador; es decir es un dispositivo electrónico que se intercala en la línea de transmisión, entre la computadora y la línea de transmisión, para transformar, adecuar y amplificar la señal digital en una señal analógica, capaz de ser transmitida por las líneas telefónicas.

El diagrama de bloques de un sistema de transmisión con modems es el siguiente:



En su estructura básica el módem está compuesto de tres grandes secciones, que son: fuente de tensión, transmisor y receptor. La fuente de tensión proporciona las tensiones adecuadas de A.C. y D.C. para la operación de la circuitería del módem.

El transmisor está compuesto de los siguientes bloques: modulador, amplificador, filtros paso banda asociados,

conformadores de pulsos y circuitos controladores de nivel.

El modulador es la parte funcional del transmisor, el cual convierte los niveles de datos de D.C. a tonos audibles de marca y espacio.

El receptor contiene un circuito demodulador, el cual proporciona las funciones inversas del transmisor; es decir, demodula los tonos audibles a niveles de datos de d.c. entendibles para la computadora o la terminal según sea el caso. Se utilizan circuitos especiales en el receptor para la amplificación, filtrado, conformación de la señal y niveles de control.

2.7.1.- Modos de operación.

Los tipos de transmisión pueden ser designados como: síncronos ó asíncronos, serie ó paralelo. Síncrono ó asíncrono se refiere al agrupamiento de los datos y tiempos. "Serie" y "Paralelo" se refiere al método para transferir la información por los circuitos de comunicación.

2.7.1.1.- Transmisión Síncrona.

Este tipo de transmisión se utiliza en sistemas en donde el reloj interno del computador central se utiliza para sincronizar al transmisor y al receptor. La señal de reloj se transmite del computador central a los dispositivos periféricos por medio de un carácter de sincronía llamado "SYN". Durante la transmisión de la información, se incluyen estos caracteres de sincronía, con lo cual los dos sistemas se sincronizan.

Una vez que la terminal receptora ha detectado el carácter de sincronía, la transmisión de datos procede carácter por carácter, sin la utilización de señales adicionales.

El dispositivo receptor acepta los datos del transmisor hasta que detecta un carácter especial de fin de transmisión con lo que se sabe que la transmisión de ese mensaje ha finalizado.

La cadena de caracteres recibida, es interpretada teniendo como base el reloj proporcionado por la sincronía. El bloque del mensaje usualmente está compuesto de 1 ó 2 caracteres de sincronía, los datos y caracteres de control, (típicamente 100 a 10,000) un carácter de terminación y 1 ó 2 caracteres de sincronía.

Esta hoja fué dejada en blanco para
propositos de edición y formato.

2.7.1.2.- Transmisión Asíncrona.

En esta transmisión no existen caracteres de sincronía durante la transmisión, en su lugar se utilizan bits de "START" o arranque y "STOP" o parada. Los cuales se incluyen en cada carácter que será transmitido. Es decir, si se desea transmitir un carácter ASCII que tiene 7 (siete) bits, se incluyen dos bits más para el arranque y la parada. Por ejemplo: transmitamos la letra A, ésta tendrá la siguiente conformación de pulsos:

```
6 5 4 3 2 1 0
1 0 0 0 0 0 1
-|-|-|-|-|-|-
```

Si ahora incluimos las señales de arranque (a), parada (s) y además el bit de paridad, suponiendo una paridad par (p), tendremos:

```
a p 6 5 4 3 2 1 0 s
1 0 1 0 0 0 0 0 1 0 0
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-
```

Si observamos la variación de la señal con respecto al tiempo, podemos ver que la señal de "STOP" o de parada siempre dejará al canal en cero (0), para que al entrar la señal de "START" o arranque, haya un cambio en el nivel de la línea de cero a uno lógico. Con esto se avisa a la interfaz que empezará la transmisión de un carácter.

Este tipo de transmisión es un poco más lenta, pues por cada carácter que se transmita, es necesario incluir las señales antes mencionadas, mientras que en la transmisión síncrona no se incluyen.

La transmisión asíncrona es adecuada cuando la transmisión de la información sucede en forma irregular, tal como la de un operador de un teclado en una terminal, en la que el intervalo entre un carácter y el siguiente es bastante grande, comparado con los ciclos de trabajo de los procesadores, es decir; el operador de una terminal, la mayor parte del tiempo no hace nada con el teclado.

2.7.1.3.- Transmisión Serie.

En este modo de transmisión los datos son enviados por un solo canal de comunicación, es decir los 7 ó ocho bits de la información que queramos transmitir, deberán ser enviados uno por uno hasta completar la transmisión de todo el carácter deseado.

2.7.1.4.- Transmisión Paralela.

Para este tipo de transmisión se utiliza un canal de comunicación para cada uno de los bits que se desean transmitir, es decir; si se transmitirá un carácter ASCII con su bit de paridad, se necesitan por lo menos 8 (ocho) líneas, más las líneas necesarias para las señales de control. Generalmente este tipo de transmisión se tiene entre el computador central y los dispositivos periféricos que estén ubicados en forma local, generalmente en impresoras, lectoras de cinta, así como lectoras de tarjetas.

En la UNAM se utiliza generalmente la transmisión en forma Síncrona y Serie, por esta razón enfocaremos el estudio de los modems sólo a los que reúnan estas características.

La siguiente figura ilustra un módem típico en diagrama de bloques y se ilustran sus funciones básicas.

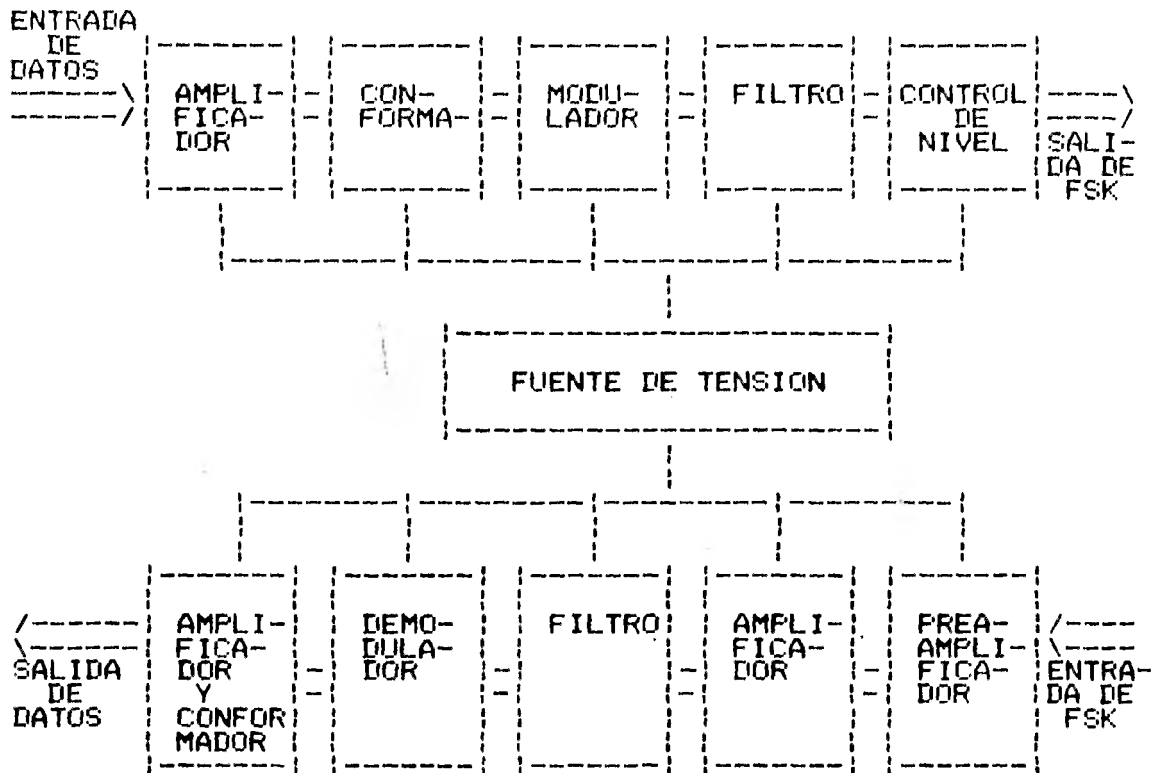


Diagrama de bloques básico de un módem.

De la figura anterior podemos ver: Los niveles de tensión de d.c. de los datos son cambiados a una forma con la cual puede operar el circuito modulador.

La salida del modulador es una señal FSK, pero tiene muchas armónicas. El multivibrador que se utiliza normalmente como oscilador de transmisión genera la frecuencia fundamental de marca y espacio, pero las señales son distorsionadas.

El filtro de la etapa de transmisión remueve estas armónicas, con esto la forma de onda a la salida es casi senoidal.

Los circuitos controladores de nivel son usualmente en etapas variables, para asegurar que los niveles de

transmisión correctos se apliquen a la línea telefónica. Las salidas típicas de los niveles de control tienen una variación de: -1, -2 ó -4 dbm en cada paso, en el rango de 0 a 16 dbm's.

La señal recibida de la línea telefónica es pequeña en amplitud, generalmente -12 a -20 dbm's. Los circuitos preamplificador y amplificador, elevan la potencia y el nivel de la señal, el filtro paso banda, restringe el paso de frecuencias de ruido fuera del ancho de banda establecido de las frecuencias de marca y espacio. Angostando el ancho de banda efectivo, se aumenta la relación Señal a Ruido (SNR).

El demodulador responde a las señales de marca y espacio produciendo los niveles de datos correspondientes para las señales digitales de la salida.

Un circuito conformador o restaurador de pulsos se utiliza generalmente para darle simetría a las señales, así mismo se les da cuadratura.

Los circuitos amplificadores cambian los niveles de d.c. a los niveles lógicos de una interfaz del tipo RS-232C.

En la práctica, muchos otros circuitos adicionales tienen los módems, se incluyen circuitos para manejar las señales supervisoras y de control. Estas señales supervisoras y de control caen en dos categorías:

(1) Aquellas que ocasionan la operación del módem.

(2) Aquellas que informan a la computadora el estado del módem o del circuito de comunicación.

La siguiente figura ilustra las señales de entrada y salida más comunes de un módem. Muchas otras se utilizan dependiendo del tipo de módem.

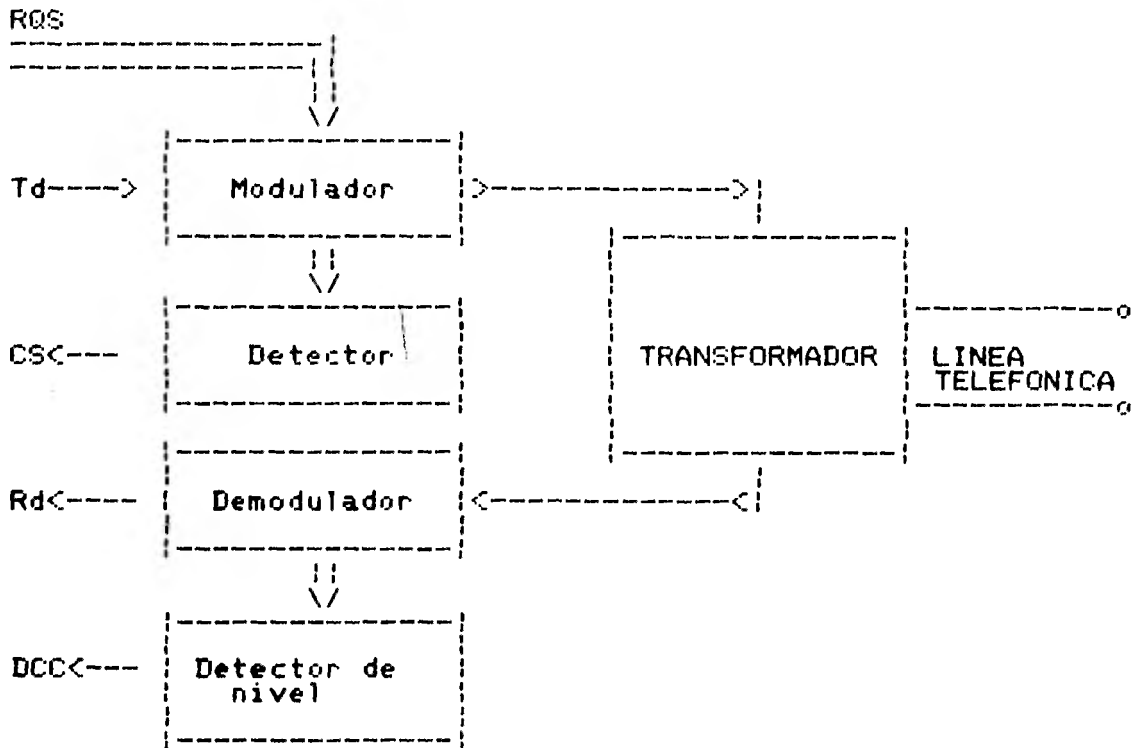


Diagrama funcional de las señales de control.

El modulador no opera hasta que la señal "Request to Send" (RQS) es proporcionada por la computadora. Este nivel de tensión habilita al transmisor a operar, proporcionando con esto una portadora a la salida (generalmente la señal de espacio).

Un circuito detector reconoce que el oscilador de transmisión está operando y genera con esto la señal llamada "Clear to Send" (CS) a la computadora. La señal "CS" informa a la computadora que el módem está listo para trabajar.

Después la computadora genera la señal llamada "Transmit Data" (Td), la cual la recibe el módem y la transforma en una señal FSK.

La señal "CS" puede ser directa o con un cierto retraso. "CS" generalmente se retarda en el orden de 200 ms para permitir a la línea telefónica y al equipo que se estabilice y permitir a la terminal receptora que se prepare para la recepción de datos.

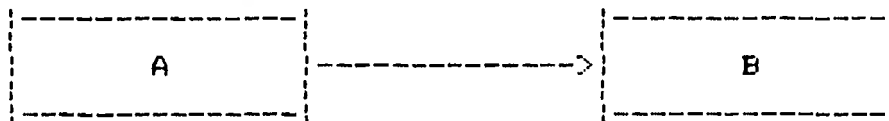
Cuando las señales FSK se comienzan a recibir de la línea telefónica, son aplicadas al circuito demodulador. Si "CS" es retardada en el modem transmisor, la portadora (2200 Hz) comienza a transmitir por un periodo de tiempo anterior a la transferencia de datos. Durante este tiempo, la presencia de la portadora en el demodulador es detectada por un detector de nivel, el que genera la señal "Data Carrier Detect" (DCD) para que la capte la computadora. La presencia de DCD en la computadora indica que la transferencia de datos está por efectuarse, y la terminal efectúa las operaciones necesarias para recibir los datos.

2.7.2.- Tipos de canales.

Un circuito de comunicación puede ser de diferentes tipos; entre ellos tenemos: Simplex, Half-duplex y Full-duplex.

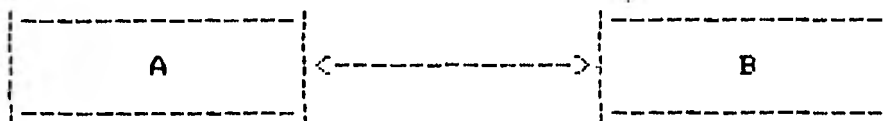
i).- Simplex.

Es posible el transmitir de A hacia B, pero no de B hacia A.



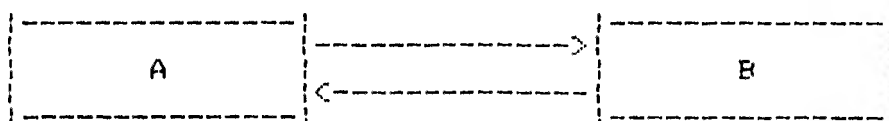
ii).- Half-duplex.

Se puede transmitir de A hacia B y de B hacia A, pero nunca simultáneamente, por ser solo uno el canal de comunicaciones.



iii).- Full-duplex.

Se puede transmitir de A hacia B y de B hacia A, y la comunicación puede ser simultánea.



2.7.3.- Ancho de banda de un canal.

Además de la dirección de transmisión, un canal se caracteriza por su ancho de banda. En general, cuanto más grande es el ancho de banda asignado para un canal, es mayor la posibilidad de obtener velocidades de transmisión mayores. Esta velocidad generalmente se mide en términos del número de señales por segundo que atraviesan un canal de comunicaciones, a esto se le conoce como razón de baudaje.

2.7.4.- Modulación.

La mayoría de los modems que se encuentran actualmente en el mercado, utilizan la modulación llamada F.S.K. En este sistema de modulación, la portadora (que opera a 1,700 Hz) es modulada con más ó menos 500 Hz para representar el 0 lógico ó el 1 lógico. Por lo que una frecuencia de 1200 Hz representa un cero, mientras que la frecuencia de 2,200 Hz representa un 1 lógico.

2.7.5.- Dispositivos Auxiliares.

Existen dispositivos que no utilizan la modulación para la transmisión de datos en líneas de transmisión, solo que estos dispositivos tienen la restricción de solo funcionar a cortas distancias, a estos dispositivos se les llama: "Line Driver".

*referencia 2.5.

CAPITULO TERCERO.
MICROPROCESADOR.

3.1.- Selección de la microcomputadora adecuada.

En la actualidad se encuentran en el mercado un gran número de microcomputadoras, algunas con un desarrollo más marcado en el área de software, pero careciendo de un sistema óptimo en hardware; algunas otras tienen un sorprendente avance en hardware, pero desgraciadamente su desarrollo en software es muy limitado. Por esto es necesario seleccionar la microcomputadora que más convenga a los intereses, políticas de desarrollo y atención al usuario de la Universidad.

Se pretende seleccionar una microcomputadora que reúna las características más sobresalientes de todas las que se encuentran en la actualidad en el mercado.

Las consideraciones ideales que requerimos de los equipos de este tipo, son: la posibilidad de proceso local y la comunicación con los equipos centrales de cómputo. En estos se encuentran los bancos de datos de la UNAM, así como los grandes sistemas de información y programas específicos de cada usuario.

Por este motivo se desarrolla la siguiente evaluación de equipos de microcomputadoras, para tratar de seleccionar la que reúna los requisitos mencionados.

3.1.1.- Consideraciones generales:

Los equipos evaluados para este estudio fueron probados en su gran mayoría revisándose aspectos tales como:

a) Posibilidades locales.

- Facilidades del sistema operativo.
- Lenguajes y software existente.
- Expandibilidad del sistema.
- Posibilidad de multi-usuario.
- Compatibilidad con otros sistemas.
- Capacidad de almacenamiento local.
- Facilidad de edición.
- Procesamiento local.

b) Posibilidades remotas.

- Comunicación como terminal.
- Transferencia y recepción de archivos.
- Facilidad de uso.

c) Características generales.

- Confiabilidad
- Modularidad.
- Presentación
- Tecnología.
- Flexibilidad.
- Tipo de compañía.

A continuación se explicarán en que consisten los requerimientos antes mencionados.

- Facilidades del sistema operativo.

Un sistema operativo nos debe brindar al menos las facilidades de un lenguaje simple de comandos para comunicación con la máquina, el manejo de información en disco, acceso a directorios y archivos que garanticen la seguridad de la información.

Sería deseable que permita la relocalización de programas en memoria, para aprovecharla al máximo brindando algún método de manejo de memoria, ya sea automático o bien que le permita al usuario definir la relocalización dentro de ésta.

Será deseable en algunos casos la facilidad de multiprogramación, manejando un sistema Multitasking que permita la expandibilidad a multiusuario. En otros casos el uso de varios sistemas idénticos será una solución más conveniente. Deberá ser un sistema operativo avanzado que explote al máximo las facilidades de la máquina y que permita al usuario hacer uso fácilmente de ellas.

- Lenguajes y Software existente.

Deberá de tener por lo menos algún lenguaje de programación que permita al usuario definir sus problemas, así como un editor y programas de servicio como:

- Copiado de archivos
- Editor de archivos
- Debusser
- Manejo de directorios

De preferencia deberá contar con lenguajes como :

Basic

Fortran

Cobol

Que son los lenguajes más comunes que nuestros usuarios manejan. Además de la posibilidad de tener algún lenguaje estructurado del tipo de Pascal.

- Expandibilidad del sistema.

Para algunas aplicaciones, el crecimiento puede ser en número de equipos, pero en otras será necesario poder expandir un equipo en tamaño, ya sea memoria, discos o terminales, ya que si en un momento dado llegan a crecer los requerimientos locales del sistema, éste se debe poder expandir hasta un límite razonable a un bajo costo.

- Posibilidad de Multi-Usuario.

Será deseable que no solamente una persona haga uso del sistema, sino que simultáneamente varias personas puedan estar editando, corriendo, creando programas o consultando una base de datos en la misma máquina y que el sistema sea capaz de hacer todo esto sin entrar en conflicto.

- Compatibilidad con otros sistemas.

En los últimos años ha habido una revolución en cuanto a equipos de cómputo se refiere y se ha visto la necesidad de que exista una compatibilidad para que algo desarrollado funcione sin muchos cambios en otro sistema de las mismas características generales. Por lo que fue creado un sistema operativo llamado CP/M (ver apéndice I) el cual trata de estandarizar muchos conceptos, de manera que los usuarios de este sistema puedan fácilmente aprender la utilización de algún otro, sin muchas complicaciones.

Es muy importante que el equipo a adquirir, cuente con este sistema operativo CP/M, para tratar de establecer un estándar dentro de la Universidad, y con esto lograr una mayor flexibilidad.

- Facilidades de edición.

La edición de programas o textos es básico en cualquier sistema, por lo tanto, este sistema deberá tener un editor poderoso, accesible al usuario y de fácil manejo que permita crear archivos y modificarlos. De preferencia deberá contar con manejo de pantalla, para una mayor rapidez en la edición.

- Procesamiento local.

Además de tener un editor, lenguajes y utilerías, el sistema debe demostrar el poder ejecutar los programas de los usuarios, haciendo uso de los puertos del equipo, el acceso a archivos sin entrar en conflicto con el sistema y brindando la posibilidad de ejecutar varios procesos simultáneos como ediciones, compilaciones y ejecuciones.

- Comunicación como terminal.

Debe ser posible que a través de este sistema se pueda establecer una comunicación con el equipo Burroughs de forma tal, que un usuario a través de la terminal del equipo se convierta en una terminal de la Burroughs, tal y como si estuviera conectada directamente, permitiéndole entrar a Cande (ver inciso 2.4) a ejecutar programas, lanzar jobs, (trabajos desarrollados en cadena en el computador central con una secuencia establecida) etc., con la ventaja de poder almacenar localmente en la Microcomputadora alguna información o transferir a la Burroughs algún archivo.

Para el enlace será necesario desarrollar el software necesario para lograr la comunicación entre ambas computadoras. La presente tesis tiene como objetivo principal el desarrollo de este software. (ver inciso 3.5). El enlace se efectuará a través de una tarjeta de hardware que es preciso seleccionar para determinar si se efectuará por interrupciones o por poleo. Hay que definir de la misma manera si esta tarjeta formará parte del hardware estandar de la microcomputadora o bien se puede añadir a la arquitectura sin ningún problema.

- Transferencia y Recepción de Archivos.

El sistema nos debe brindar la posibilidad de mandar y recibir archivos a través de una línea telefónica comunicada a la Burroughs. En un principio esta transferencia será del tipo Contention en ASCII con protocolo TTY o sea sin ningún chequeo, pero deberá ser confiable y tal que nos permita por lo menos los 1200 bauds que es la velocidad de transferencia con módem, y además que ofrezca la posibilidad de transmisión hasta 4800 y 9600 bauds.

-Facilidad de uso.

Tanto el entrar en modo de terminal; el transferir y recibir archivos, deberá de ser de una forma simple a través de comandos accesibles al usuario que le permitan fácilmente hacer lo que éste desee (ver inciso 3.5.5).

- Confiabilidad.

El equipo deberá ser confiable, o sea que no se descomponga fácilmente, lo que significa que sea un equipo de

buena calidad y diseño.

- Modularidad.

Debido a que siempre existe la posibilidad de falla, en un diseño modular es posible el corregir fácilmente una falla cambiando el módulo descompuesto, lo que hace que el servicio al usuario sea casi inmediato. Estos módulos se pueden tener en un inventario de refacciones, además de que en el laboratorio pueden ser corregidos. Otra ventaja de la modularidad es la facilidad de crecimiento.

- Presentación .

Otro punto muy importante es la presentación. Por medio del cual se evitarán posibles fallas, debido al mal manejo de este. Es decir, un equipo del tipo "desarrollo" generalmente viene en una sola tarjeta sin cubierta exterior, lo que podría ocasionar desperfectos constantes.

- Tecnología.

Dados los nuevos avances tecnológicos en el área y la velocidad en que dan los eventos en nuevos sistemas y equipos, es recomendable que se adquiera un equipo que conjunte en él los últimos avances tecnológicos, que se traducen en una alta integración, un bajo consumo de energía, bajo costo, alta confiabilidad y alta capacidad de procesamiento.

Con los nuevos microprocesadores sería deseable un equipo que hiciera uso de estos tipos de circuitos especializados en cada parte dándole una mayor potencialidad.

- Flexibilidad.

Nos debe permitir el crecimiento, tanto en capacidad de almacenamiento, como en manejo de dispositivos, como Impresoras y en su caso otro tipo de dispositivos especiales según la aplicación específica de cada usuario.

- Tipo de compañía.

En la actualidad existe un gran número de compañías que fabrican microcomputadoras; la compañía seleccionada deberá ser una que esté perfectamente establecida en el mercado para asegurar con esto la calidad del equipo y la supervivencia de ésta compañía en el mercado a través de los años, para que pueda brindar un soporte en todas las áreas que se requiera.

- Equipos considerados.

A continuación se da una lista de los equipos considerados en el presente estudio.

Adds
Altos
Apple II
Archives
Atari
Cromemco
Codex
Heath Kit
Hewlett Packard
Intel
Radio Shack
Super Brain
Televideo
Tektronix
Texas Instruments

* referencia 3.12

3.1.2.- Datos de Software y Hardware.

A continuación se dan dos tablas comparativas de las características sobresalientes, la primera sobre datos de Hardware y la segunda sobre datos de Software.

DATOS DE HARDWARE

Fabric.	Modelo	C P U			RAM		Floppy	
		\$\$	Nombre	Bits	Bus	STD		Max
TANDY	TR80 II	3,899	Z80	8	PROPIO	64K	64K	.486M
APPLE	APLEE II	2,500	6502	8	PROPIO	16K	48K	.116M
ATARI	800	2,000	6502	8	PROPIO	16K	48K	.3MB
ALTOS	ACS-8000-5	5,990	Z80A	8	No	64K	208K	1MB
ADDS	MULTIVISION1	3,785	8085A	8	PROPIO	64K	256K	.78MB
VECTOR	VIP	3,695	Z80A	8	S-100	56K	224K	.315M
CROMEMCO	SISTEMA 0	5,500	Z80A	8	S-100	64K	64K	.76MB
CROMEMCO	SISTEMA 2H	9,995	Z80A	8	S-100	64K	512K	.76MB
ONTEL	OP1/50	7,800	8086	8	PROPIO	64K	64K	1MB
INTERFACE	SUPER BRAIN	2,500	Z80	8	S-100	32K	64K	.35MB
HEATHKIT	WH89-CA	2,895	Z80	8	PROPIO	48K	64K	.1MB
ARCHIVES	ARCHIVES	4,800	Z80	8	PROPIO	64K	64K	.7MB
TELEVIDEO	SYSTEM 1	3,995	Z80	8	PROPIO	64K	64K	1MB
CODEX	CSX68/1	3,975	6800	8	PROPIO	32K	64K	1MB
TEXAS					DP			
INSTRUMENT	DS990/1	9,995	TMS9900	16	PROPIO	32K	32K	NO
HEWLET								
PACKARD	85	3,700	HP	8	GPIB	16K	32K	NO
TEKTRONIX	4051	6,900	TEK	8	GPIB	16K	64K	

nota!

*Los precios estan dados en U.S. Dollar

*Todos estos sistemas incluyen el Precio de terminal.

* Datos de enero de 1981 (referencia 3.1)

DATOS DE SOFTWARE

Marca	Modelo	Sist.		Operat.					Leng. de Progr.				
		Mono U.	Multi U.	A	B	C	F	P	A	B	C	F	P
Tandy	TRS 80 Mod II	TRS2DOS	no	si	si	OP	OP	no					
Apple	Apple II	---	no	si	si	OP	OP	si					
Atari	800	---	no	si	si	no	no	no					
Altos	ACS- 8000-5	CP/M	MP/M	si	OP	OP	OP	OP					
ADDS	Multi- vision I	Muon	Muon	si	si	OP	OP	si					
Vector	VIP	CP/M	no	si	si	OP	OP	OP					
Cromemco	2 ó 3	CDOS	Cromix	si	si	si	si	OP					
Intel	Op 1	Dos	si	si	si	si	OP	OP					
Intertec	Super Brain	CP/M	no	si	si	si	OP	OP					
Heath	Wh 89	OP	no	OP	OP	OP	OP	OP					
Archives	archi- ves	CP/M	no	si	si	OP	OP	OP					
Televideo	System I	CP/M	Mmmost	si	si	si	OP	OP					
Codex	CDX	si	no	si	si	si	OP	OP					
Texas Instrument	DMS990-1	TMS	no	si	si	no	no	no					
Hewlett Packard	85	Hp	no	si	si	no	no	no					
Tektroni	4051	Tek		si	si	no	no	no					

donde:

A = Assembler

B = Basic

C = Cobol

F = Fortran

P = Pascal

OP = opcional (no lo soporta el fabricante)

no = no lo tiene

si = sí lo tiene (lo soporta el fabricante)

nota:

* Datos de enero de 1981 (referencia 3.1)

3.1.3.- Consideraciones Particulares.

Para cada caso en especial, de los descritos a continuación vamos a ver cuales son las necesidades en cuanto a equipo de cómputo, así como las posibilidades o alternativas de solución.

-Apoyo administrativo

Dentro de esta área tenemos :

- Substitución de tarjetas perforadas
- Apoyo a proceso de control administrativo tales como:

- a)Inscripciones
- b)Control de personal
- c)Presupuesto
- d)Inventario

- Apoyo al desarrollo de nuevos sistemas.
- Procesamiento de la palabra

A continuación se explicarán cada una de las necesidades:

- Substitución de tarjetas perforadas

En este caso se requiere un equipo que permita el almacenamiento de información. Con tener el equivalente a la capacidad de 1,000 tarjetas puede ser bastante bueno. Se requiere un teclado para meter esta información y una vez capturada, debe haber un lugar en donde pueda ser transferida

al equipo Burroughs (lectora de floppies). Para esto se requiere del software necesario para que simule a una perforadora de tarjetas. En algunos casos, será deseable el poder usar varias terminales (hasta 4), ya que este proceso no consume grandes recursos de procesador y el mismo equipo pudiera hacerlo. Para esto será necesario el contar con una lectora de floppies que se realizará con el software desarrollado en la presente tesis (ver inciso 3.5.2).

Para resolver esta aplicación se desarrolló la tesis llamada: "Desarrollo de un sistema para la creación y validación de bancos de información".

* referencia 3.11

- Apoyo a procesos de control administrativo.

Para estos casos se debe tener la posibilidad de almacenamiento local, de uno o varios archivos que puedan estar en floppy disk, y de preferencia contar con un sistema de bases de datos.

En algunos casos la cantidad de información se incrementa grandemente, creándose grandes bancos de datos con información que pueden ser materias, grupos, alumnos, contabilidad, presupuesto y almacén. En estos casos se requiere una mayor capacidad y la posibilidad de acceso simultáneo a través de más terminales.

El sistema a considerar será uno que permita el almacenamiento de un volumen suficiente de información, (10 MB) y la posibilidad de uno a cuatro usuarios simultáneos. Hay que considerar que para algunos sistemas, un equipo del tipo de sustitución de tarjetas perforadas en donde hay poco volumen de información puede ser conveniente. En todos estos casos es muy importante la comunicación con el equipo Burroughs para la integración al banco de datos generales de cada sistema.

- Apoyo al desarrollo de nuevos sistemas.

En este caso, se requiere de un equipo que permita localmente la edición de programas, el poder probar aquellas rutinas ya programadas, poder copiar programas y comunicarse a la Burroughs; por lo que un sistema con una capacidad local de almacenamiento de 500 K puede ser satisfactorio.

En un sistema de este tipo, podría ser deseable que permitiera mayor almacenamiento, además de la posibilidad de que varios usuarios lo compartan simultáneamente.

- Procesamiento de palabra.

Este caso es similar al caso anterior, pero se requiere la posibilidad de tener una impresora de preferencia del tipo Daisy-Wheel, (Sistema intercambiable de tipos de letras, para lograr calidades en la impresión como las máquinas de escribir) por lo que un sistema multiusuario en el que haya colateralmente una impresora compartida, es el más indicado.

- Investigación

En este caso requerimos un sistema del tipo de apoyo al desarrollo de nuevos sistemas, que permita la creación de programas, la compilación y prueba en forma local, además de la protección de archivos en forma local, en unidades de floppy disk, así como el procesamiento de palabras.

También es deseable el poder contar con tarjetas de electrónica tales como:

- Convertidores Analógico/Digital y Digital/Analógico
- Controladores
- Interfases Serie/Paralelo y Paralelo/Serie

- Docencia.

Para el caso de la docencia, tenemos equipo para sustitución de tarjetas ya mencionados y también se ve la conveniencia de equipos de un solo usuario, en que el alumno tenga a su disposición un equipo completo para aprender el uso de la máquina y hacer lo que desee sin la necesidad del uso de grandes volúmenes de información.

A continuación se presentan tablas comparativas para cada una de las aplicaciones antes mencionadas.

Apoyo a procesos de control Administrativo.

Sistema	1 Usuario			4 Usuarios		
	Memoria	Disco	Precio	Mem.	Disco	Cant. Precio
Altos	208 k	10 MB	12,000	208K	10 MB	1 15,500
ADDS	64 k	10 MB	8,595	256K	10 MB	1 16,085
Vector	64 k	10 MB	8,659	256K	10 MB	1 -----
Cro-						
memco	64 k	10 MB	11,800	256K	10 MB	1 20,060
Intel	64 k	10 MB	13,800	256K	10 MB	1 32,000
Tele-						
video	64 k	10 MB	8,995	320K	10 MB	1 14,380

* Datos de enero de 1981 (Referencia 3.1)

Substitución de perforación, apoyo al desarrollo de nuevos sistemas e investigación.

Sistema	1 Usuario			4 Usuarios			Precio
	Memoria	Disco	Precio	Mem.	Disco	Cant.	
Radio-Shack	64 k	500KB	3,500	256k	2 MB	4	14,000
Apple	48 k	500KB	3,000	192k	2 MB	4	12,000
Altos	208 k	1 MB	7,000	208k	2 MB	1	10,500
AIDS	64 k	700KB	4,100	256k	700k	1	11,200
Vector	64 k	1 MB	3,695	256k	1 MB	1	-----
Cro-memco	64 k	1 MB	5,200	256k	1 MB	1	14,700
*Intel	64 k	1 MB	7,800	256k	1 MB	1	27,000
Super Brain	64 k	700KB	2,500	256k	2.8MB	4	10,000
Heath Kit	48 k	400KB	2,895	192k	1.6MB	4	20,000
Archi-ves	64 k	700KB	4,800	256k	2.8MB	4	19,200
*Codex	64 k	1 MB	9,700	256k	2 MB	4	34,000
Tele-video	64 k	2 MB	3,975	520k	11 MB	1	14,390
*HP	64 k	Cinta	3,700	64k	.8MB	4	15,000
*TI	32 k	Cinta	4,995	256k	2 MB	4	20,000
*Tek-tronix	16 k	Cinta	6,900	64k	1 MB	4	28,000

notas:

- * No se aplica a substitución de perforación.
- * Precios en U.S. Dollar
- * Datos de enero de 1981 (Referencia 3.1)

Procesamiento de palabra

Nota: Agregar a cada sistema de 2,000 a 4,000 U.S. por la Impresora.

Sistema	1 Usuario		Precio	4 Usuarios		Precio
	Memoria	Disco		Memoria	Disco	
Radio-						
Shack	64 k	446 k	3,899	NO	NO	NO
Apple	48 k	446 k	3,000	NO	NO	NO
Altos	208 k	1 MB	5,990	208 k	2 MB	10,500
ADDS	64 k	1.2 MB	3,875	256 k	700 k	11,200
Vector	64 k	315 k	3,695	256 k	1 MB	-----
Cro-						
memco	64 k	1.2MB	5,500	256 k	1 MB	14,700
Ontel	64 k	1 MB	7,800	256 k	1 MB	27,000
Super						
Brain	64 k	700 k	2,500	NO	NO	NO
Heath-						
Kit	48 k	400 k	2,895	NO	NO	NO
Archi-						
ves	64 k	700 k	4,800	NO	NO	NO
Codex	64 k	1.2MB	2,895	320 k	11 MB	14,380
Tele-						
video	64 k	2 MB	3,975	320 k	11 MB	14,380

notas:

*Precios en U.S. Dollar

*Datos de enero de 1981 (referencia 3.1)

Docencia

Sistema	Memoria	Disco	Precio
Radio Shack	64 k	446 kb	3,889
Apple	48 k	446 kb	3,000
Altos	208 k	510 kb	5,990
ADDS	64 k	1.2 mb	4,100
Vector	64 k	315 kb	3,695
Cromemco	64 k	760 kb	5,500
Super Brain	64 k	700 kb	2,500
Heathkit	48 k	100 kb	5,000
Archives	64 k	700 kb	4,800
Codex	64 k	1.2 mb	3,975
Televideo	64 k	1.2 mb	3,995

notas:

* Los precios son en U.S. Dollar

* Datos de enero de 1981 (referencia 3.1)

A continuación se muestra una lista con las pruebas que se ejecutaron en cada uno de los equipos bajo estudio.

1.- Funcionamiento del sistema operativo.

a).- Creación de archivos en disco.

b).- Edición de archivos.

c).- Verificación del contenido de los archivos creados.

d).- Ejecución de programas.

e).- Realizar archivos de comandos ejecutables.

2.- Uso del equipo.

a).- Como terminal.

i).- Velocidad de transferencia.

ii).- Bits de stop y paridad (en caso asíncrono).

iii).- Modo de transmisión (Half duplex, Full duplex, síncrona, asíncrona).

iv).- Número de puertos.

v).- Compatibilidad con la interface RS-232C.

b).- Como terminal inteligente.

i).- Transferencia de archivos de floppy a la computadora central.

ii).- Transferencia y almacenamiento de archivos de la computadora central al floppy.

iii).- Programa terminal.

c).- Facilidades.

i).- Teclas programables.

ii).- Implementación del BREAK.

3.1.4.- Conclusiones.

Hay que considerar que las necesidades de la UNAM en esta área pueden ser para fines prácticos casi ilimitadas, por el gran volumen de aplicaciones potenciales; esto lo demuestran estadísticas de uso de los equipos con que cuenta el C.S.C. Considerando que sólo una parte muy pequeña de la comunidad Universitaria está haciendo uso de estos recursos.

En el caso de captura de datos, la Universidad cuenta con gran cantidad de equipos de perforación; la conveniencia de cambiarlos resulta evidente, debido a la relación de los costos actuales de estos, con las microcomputadoras, esto sin considerar el uso de los dispositivos de Floppy Disk. Como otra característica, podemos añadir que el floppy es re-utilizable, reduciendo con esto notablemente los costos en comparación con el sistema de tarjetas tradicional.

El caso de procesamiento de palabra es una aplicación potencial muy amplia, ya que en cada dependencia universitaria podría existir un equipo de este tipo.

Dependiendo del volumen o requerimientos podría ser aún mayor.

En caso de apoyo a procesos administrativos, investigación y desarrollo de nuevos sistemas, la escala es menor, ya que se reduce primordialmente al crecimiento requerido de algunos usuarios del Centro de Cómputo, como nuevas dependencias que actualmente se encuentran en el proceso de inicio de automatización.

Con estas consideraciones se requiere que el equipo que sea introducido en la UNAM, no sea un equipo experimental. Además se requiere que sea un equipo que lleve algún

tiempo en el mercado, demostrando su confiabilidad, calidad y desarrollo en software, para que la introducción sea exitosa y no cause problemas en un principio. Deberá ser un equipo que no requiera de expertos para manejarlo y que se adapte en mayor grado a las condiciones generales mencionadas al principio de este documento.

De la lista de equipos considerada y de la experiencia obtenida en las pruebas de estos equipos resaltan los siguientes.

ADDS

Altos

Cromemco

Radio Shack Mod. II

Ontel

Televideo

A continuación se ven cuales son los equipos probables en cada aplicación en particular .

RESUMEN DE EQUIPOS RECOMENDADOS Y EN DONDE SE APLICAN

Sistema	Apoyo substi- tución Investi- sación	Docen- cia	Admi- nistra- ción	Procesa- miento de Palabra	# de Aplica- ciones
ADDS	Si	No	No	No	1
Tele- video	Si	Si	Si	Si	4
Altos	Si	No	Si	Si	3
Cro- memco	Si	Si	Si	Si	4
Radio- Shack	Si	Si	No	No	2
Ontel	No	No	Si	No	1

ADDS.- Fue probado y aunque en un principio resultó atractivo, después se encontraron pequeños problemas tales como: el software no funcionaba satisfactoriamente. En el caso del multiusuario, no funcionó adecuadamente. El software con que cuenta el sistema es muy limitado, pues solo es el editor. Además de no contar con el tan popular y conveniente Bus S-100.

Intel. Resulta ser el equipo más costoso. Es un equipo orientado principalmente al proceso administrativo y no cuenta con una gran cantidad de software; Tampoco posee el Bus S-100.

Televideo.- En este caso se trata de un nuevo equipo que acaba de salir al mercado. No cuenta con el Bus S-100. Y no se tiene gran experiencia en este equipo por lo nuevo.

Por lo que podemos concluir que nos quedan los siguientes equipos:

Altos.- Por su flexibilidad y bajo costo en multiusuario.

Cromemco.- Por su calidad, expandibilidad, software existente y arquitectura (utiliza el Bus S-100 y CP/M).

Radio Shack.- Para aplicaciones de docencia y utilización en un usuario.

De las tres opciones anteriores debemos destacar las siguientes características deseables:

- Sistema operativo compatible con CP/M.
- Posibilidad de expansión en forma modular.
- Posibilidad de multi-usuarios.
- Bus S-100.

Debido a que la única máquina que reúne todas estas características, y es una de las más difundidas, se seleccionó a CROMEMCO, como el microprocesador que adquirirá la UNAM para todas las aplicaciones antes mencionadas.

3.2.- Descripción del sistema.

El sistema CROMEMCO es un sistema basado en un microprocesador de 8 bits, el Z80A (ver apéndice III) ; cuenta con gran cantidad de tarjetas de Hardware para su completo desarrollo; entre ellas tiene:

- Procesador Z80A
- Memoria de 64KB
- Controlador de discos.
- Controlador de impresora.
- Interfaz Serie-Paralelo, Paralelo-Serie.
- Convertidor analógico/digital y digital/análogo.
- Grabador de EPROM
- Computador en una sola tarjeta.
- Procesador de Entrada/Salida.
- Puertos de Entrada/Salida.
- Disco duro de alta velocidad.
- Monitor de color
- Tableta de graficación.

Todas las tarjetas anteriores son compatibles con el Bus S-100 (ver apéndice II); y tiene una gran variedad de software, entre estos podemos mencionar:

- Basic
- C
- Cobol
- Fortran
- Fortran Racionalizado
- Lisp
- Macro Ensamblador
- Procesador de palabras
- Base de Datos
- Graficación.
- Editores simples y de pantalla.

Todo este software es compatible con CP/M (ver apendice I). Además este sistema de computación cuenta con la capacidad de multiusuarios.

Este sistema tiene una gran expansión en la actualidad, pues son más los periféricos, así como la paquetería que cada día sale al mercado para esta máquina.

Para fines de 1982 está anunciado el poder trabajar con un procesador de 16 bits (68000). Para esto es necesario adicionar una tarjeta de hardware que contará con el Z80 y el 68000, una tarjeta de memoria de 256 KB RAM y el sistema operativo correspondiente.

En general la arquitectura mínima del sistema CROMEMCO es la siguiente:

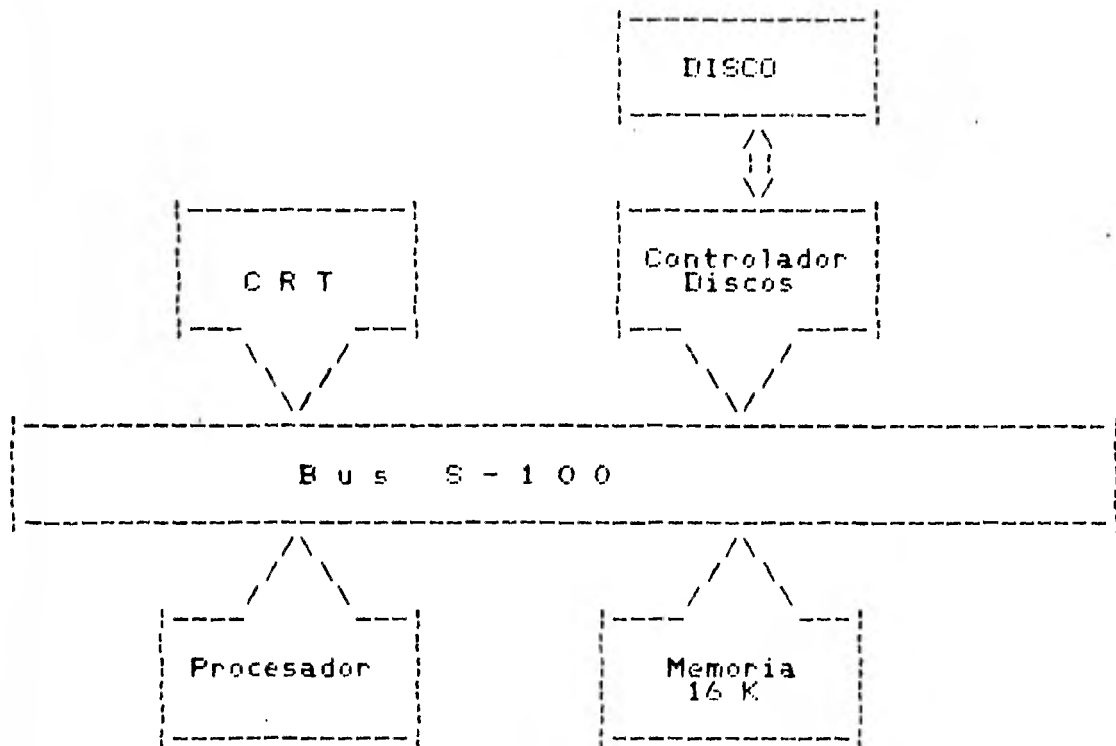


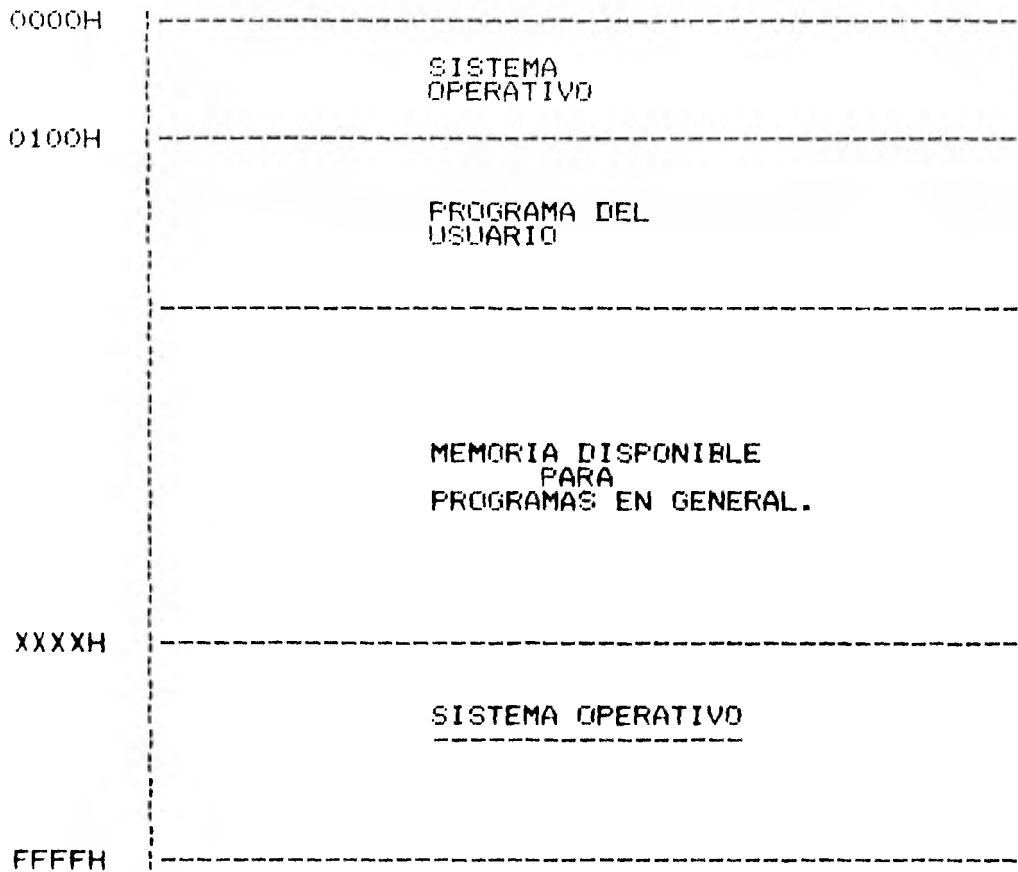
Diagrama de la arquitectura mínima del sistema CROMEMCO.

Con esta configuración básica es posible desarrollar los programas de aplicación que se quieran, pues se tiene todo lo necesario.

Es posible expandir la memoria hasta 64 K, implementar un controlador de impresora, discos de mayor capacidad y velocidad, y que varios usuarios compartan los recursos de la máquina simultáneamente (para esto es necesario la adición de algunas tarjetas de hardware y un sistema operativo especial

llamado CROMIX).

La memoria del sistema CROMEMCO está dividida de la siguiente forma, entre el sistema operativo y el usuario:



Mapa de memoria del Sistema.

Para mayor información de la estructura del Sistema Operativo de esta máquina, consultar inciso 3.4.1 y apéndice I de esta tesis.

3.3.- Hardware utilizado.

A continuación describiremos en forma general cada una de las tarjetas que componen la arquitectura básica de este sistema.

3.3.1.- Módulos del sistema.

Primeramente describiremos las tarjetas de propósito general, y posteriormente la tarjeta de comunicación de datos.

3.3.1.1.- ZPU. Unidad central de procesamiento con Z-80.

- Procesador.

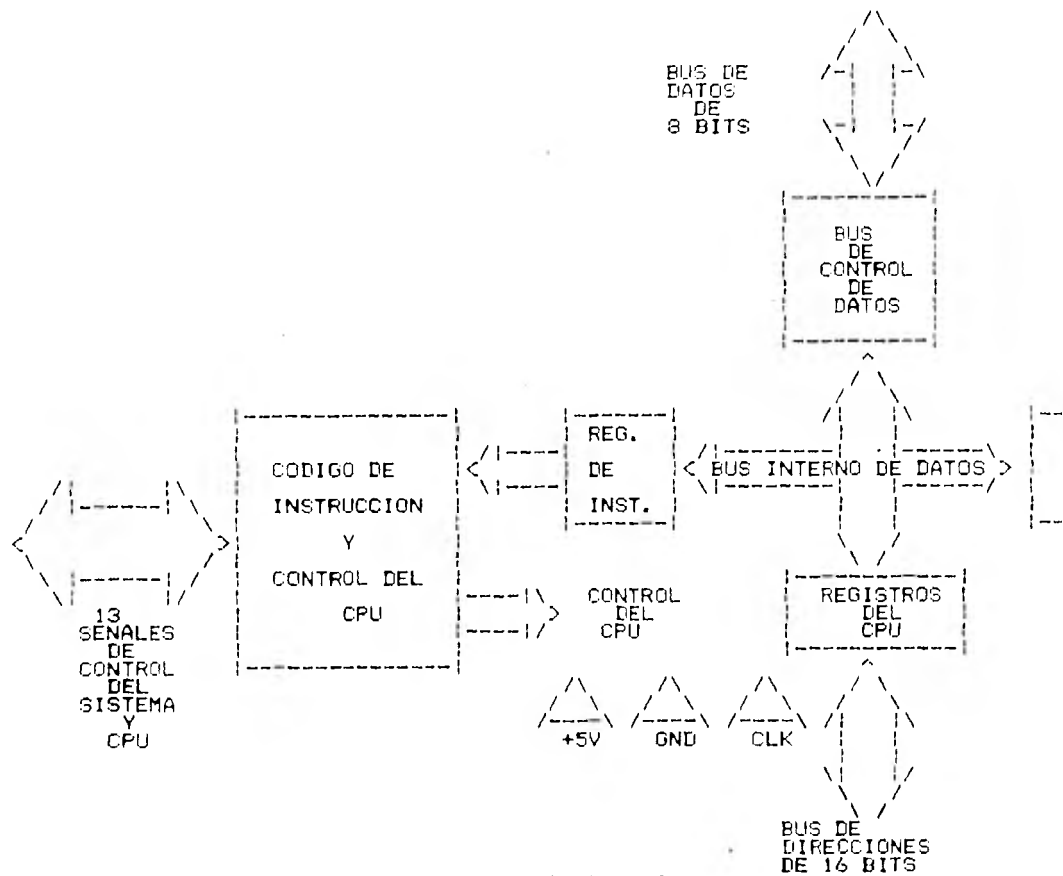
Se utiliza el Z-80A, que es una versión del mismo, pero es capaz de utilizar un reloj de 4MHZ.

Este procesador tiene las siguientes características:

Registros de 8 bytes:	16
Registros de 16 bytes:	10
Registros de índice:	2
Modos de dirección:	10
Dirección de E/S:	256
Bits de bandera:	6
Voltaje requerido:	+5V
Velocidad de reloj máxima:	4MHZ
Compatible con TTL :	SI
Modos de interrupción :	4
Refrescamiento de memoria automático:	SI
Velocidad relativa en ejecución:	1.0

- Arquitectura del CPU.

Podemos representar la arquitectura interna del CPU a través de un diagrama de bloques, como el que se muestra a continuación:



Arquitectura del CPU Z80

- Especificaciones técnicas de la tarjeta del ZPU.

Frecuencia del reloj:	Versión en 4MHz del Z-80A
Conjunto de Instrucciones:.....	158 instrucciones incluyendo las 78 instrucciones del procesador 8080.
Salto automático al encendido:	Habilitado por Hardware.
Direcciones posibles al salto automático:	16 direcciones seleccionables por switches.
Generación de estados de espera:.....	0-4 estados de espera seleccionables por Hardware.
Ciclo de espera de máquina:	Seleccionables por Hardware.
Compatibilidad de BUS:	S-100
Necesidades de alimentación:	+8 V DC y 1.1 Amp.
Rango de operación:	0-55 grados Centígrados.

- Salto automático de memoria al encendido.

El circuito de salto automático instalado en la tarjeta del ZPU, permite a ésta el ser usada en un sistema de BUS

S-100 sin controles en los paneles externos de la microcomputadora.

Cuando el sistema es encendido, el hardware del ZPU fuerza un salto automático a una de la 16 posibles localidades de memoria disponible, seleccionables por el switch de cuatro posiciones.

En la dirección que se selccione con estos switches, se encuentra el sistema operativo residente en ROM (RDOS).

El salto automático a las direcciones de memoria correspondiente a cada codificación de los switches, se tabula a continuación:

A15	SWITCH A14	A13	A12	Cuando se enciende SALTA A:
0	0	0	0	0000H
0	0	0	1	1000H
0	0	1	0	2000H
0	0	1	1	3000H
0	1	0	0	4000H
0	1	0	1	5000H
0	1	1	0	6000H
0	1	1	1	7000H
1	0	0	0	8000H
1	0	0	1	9000H
1	0	1	0	A000H
1	0	1	1	B000H
1	1	0	0	C000H
1	1	0	1	D000H
1	1	1	0	E000H
1	1	1	1	F000H

Cuando se carga el sistema operativo a la memoria, una señal llamada "Phantom" (ver apéndice II), deshabilita la memoria ROM (en donde está RDDS) para tener 64K de memoria RAM (65,536 bytes de memoria de acceso aleatorio).

- Selección de frecuencia del reloj.

El microprocesador Z-80A puede trabajar con un reloj de 4 MHz (con un ciclo de tiempo de 250 nses.) ó 2 MHz (con un ciclo de tiempo de 500 nses.) . La frecuencia de operación es seleccionable por medio de un switch tipo "toggle" . La línea del "bus" S-100 previamente etiquetada como "STCK" es utilizada por el ZPU como una línea indicadora de 4 MHz.

- Selección de estados de espera.

El ZPU provee un generador de estados de espera en la misma tarjeta, de tal manera que son compatibles la frecuencia del reloj del Z-80A y el tiempo de acceso al sistema de memoria que se tenga.

El ZPU permite la inserción de estados de espera durante cada ciclo de máquina.

- Selección de memoria espejo.

El microprocesador 8080 permite ó espejea la dirección de los 8 bits del puerto de Entrada/Salida en el orden alto y bajo de la dirección del BUS. A pesar de que esta característica no pertenece al procesador Z-80A, la tarjeta del ZPU está diseñada para imitar esta característica a través de una circuitería especial, asegurando que será compatible con un sistema que se actualice de 8080 al Z-80A.

- Refrescamiento de memoria.

Todos los tipos de tarjetas de memoria dinámica requieren un refrescamiento de las direcciones proporcionadas por el Z-80A, reflejadas en las líneas de las direcciones de orden alto.

* Referencia 3.2

3.3.1.2.- 16FDC

El 16FDC es un controlador de discos de la segunda generación que lanza al mercado CROMEMCO. Esta tarjeta está provista de un sistema completo para la operación de los floppies, que pueden ser indistintamente de 8 pulsadas y 5 pulsadas. Incluye un puerto serie de Entrada/Salida (I/O) para una terminal de video con interfaz RS-232, una memoria pre-programada ROM (Read Only Memory), en donde se encuentra grabado un programa monitor llamado RDOS (Resident Disk Operating System), por medio del cual se inicia el sistema (boot), así mismo posee comandos por medio de los cuales se realiza un diagnóstico básico de la memoria y drives. Este monitor normalmente funciona solo cuando se enciende la máquina, ó despues de un reset (ver 3.3.1.1 direcciones posibles de salto al encendido). Con este monitor se inicia el sistema y en este momento se desconecta el monitor transfiriéndose el control al sistema operativo (ver apendice I y II).

El 16FDC normalmente maneja hasta 4 drives (el drive es el dispositivo físico en donde se pone el disco para su lectura y/o escritura) en una cadena de prioridades; sin embargo, hasta 16 drives pueden ser encadenados si es necesario. (si el drive es capaz de decodificar las direcciones de las líneas).

* Referencia 3.3

3.3.1.3.- 64KZ

La tarjeta CROMEMCO 64KZ es una memoria dinámica para Lectura/Escritura de 65,536 bytes (64K bytes), compatible con el bus S-100. Esta tarjeta incorpora una memoria dinámica RAM (Random Access Memory), para esto utiliza un circuito integrado llamado: TMS 4116-15 (16K x 1 bit). Opera en un tiempo de acceso máximo de 250 nsec. Esto significa que la 64KZ puede operar en los sistemas Z80 con un reloj de 4MHz sin utilizar estados de espera en lo absoluto. Esta tarjeta maneja la señal de selección de banco, es posible el aumentar lógicamente la memoria, pues se puede intercambiar bancos de memoria por software, esta es una opción muy importante, pues se utiliza grandemente en sistemas operativos para multiusuarios.

Esta tarjeta ofrece las siguientes características:

- 64K bytes de memoria para Lectura/Escritura (R/W) en una sola tarjeta de memoria para el bus S-100.
- Un tiempo máximo de acceso de 250 nanosegundos.
- Compatibilidad con el Z80 y el 8080 CPU.
- Organización en dos bancos independientes de memoria (A y B).
- Selección del banco, permitiendo una expansión de la memoria más allá de los 64K bytes.

Especificaciones técnicas:

Capacidad de Memoria:	65,536 BYTES (64K BYTES)
Tiempo de acceso a memoria:	250 Nanosegundos (maximo)
Estados de espera a 2Mhz:	No se Requieren
Estados de espera a 4Mhz:	No se Requieren
Compatibilidad con bus:	S-100
Requerimientos de potencia:	+ 8 volts y 1.8 amperes (max) +18 volts y 0.45 amperes (max) -18 volts y 0.03 amperes (max)
Temperatura de operación:	0 - 55 Grados Celsius
Tipo de memoria:	TMS 4116-15,16x1 RAM dinámico

* Referencia 3.4

3.3.2.- TU-ART

La tarjeta TU-ART (Twin Universal Asynchronous Receiver and Transmitter ó Receptor y Transmisor Universal Asíncrono Doble) provee dos canales duplex serie, para el intercambio de información; dos canales de intercambio de datos en paralelo; y diez contadores de tiempo (Timers). Existe un puerto por el cual se le puede dar una cantidad al contador deseado. Esta cantidad es decrementada en 1 cada 64 microsegundos a partir de la carga inicial. Cuando la cuenta llega a cero, se produce una petición de interrupción y el contador es deshabilitado. Como la cuenta máxima es de 255, el intervalo de tiempo máximo que puede contar un timer es: 254×64 microsegundos = 16.32 milisegundos. La información del estado del dispositivo se puede obtener por interrupciones ó bien preguntando periódicamente (Polling). Dos líneas de petición de interrupción están disponibles para el usuario, a través de los puertos serie y/o paralelo. El TU-ART tiene su propio reloj controlado por un cristal de cuarzo, y la comunicación con el bus S-100 se realiza asincrónicamente, así que la frecuencia del reloj del CPU no es crítica, es decir el CPU se puede operar a 2 MHz ó bien a 4 MHz sin que esto afecte la transmisión de información entre el CPU y el TU-ART. El TU-ART incorpora dos circuitos integrados, controladores de los puertos del número TMS 5501 NMOS I/O.

EL CPU normalmente "ve" al TU-ART como un puerto dual de Entrada/Salida con capacidad de interrupción. El CPU normalmente lee los datos de esta tarjeta a través del bus S-100 ejecutando para ello una instrucción como la siguiente:

IN A, (PUERTO)

Existen catorce puertos utilizados para la Entrada/Salida de comandados por uno de los TMS 5501 y otros catorce puertos por el otro dispositivo (TMS 5501). Es posible el intercambiar por medio de software los TMS 5501 (el dispositivo A será el B y lo contrario). La dirección base de ambos dispositivos (A y B) es seleccionada por unos micro-switches que se encuentran en esta tarjeta. Toda la información entra o sale del TU-ART a través del acumulador, y por medio de los registros que tiene cada puerto.

Esta es una tarjeta muy importante en el estudio de esta tesis, pues es en ésta donde se realiza la conexión física con BURROUGHS. En el inciso 3.5.1 se explica con más detalle como se realiza la interconexión entre ambas computadoras.

3.3.2.1.- Velocidades

El registro 00 de salida (Out) es el registro para la selección de la velocidad. El diagrama de bits es el siguiente:

D7	D6	D5	D4	D3	D2	D1	D0
Bits de STOP	9600	4800	2400	1200	300	150	110

donde:

D7 STOP

Un "uno" en el bit 7 selecciona un bit de stop para la recepción y transmisión serie. Un "cero" en este bit selecciona dos bits de STOP.

D6 - D0 VELOCIDAD

Un "uno" en alguno de los posibles bits selecciona la correspondiente velocidad. Si por equivocación se selecciona mas de un bit, se tomará en cuenta el bits más significativo. Por ejemplo, si queremos transmitir a 4800 bits/sec y 1 bit de stop, tendremos:

1 0 1 0 0 0 0 0
A 0

Para fijar esta velocidad, deberemos de ejecutar la siguiente instrucción:

LD A,0A0H

OUT (PUERTO), A

3.3.2.1.- Interrupciones.

A la señal de control que rompe la secuencia de las instrucciones que se están ejecutando en el procesador, se le llama INTERRUPCION. Una interrupción debe ser atendida por una rutina de servicio expresamente escrita para este propósito. La secuencia que se sigue al recibir una interrupción es la siguiente:

- a).- El procesador se encuentra ejecutando las instrucciones regulares de un Programa.
- b).- Un dispositivo hace la petición de interrupción.
- c).- Si las interrupciones están habilitadas, se toma en cuenta ésta, en caso contrario se descarta la petición de interrupción.
- d).- El Program Counter se salva en el stack.
- e).- Del registro "I" se toma la dirección donde se encuentra la rutina de atención a la interrupción. El complemento de la dirección lo proporciona el dispositivo que interrumpió.
- f).- Se deshabilitan las interrupciones (si así se desea) para evitar conflictos con otros dispositivos. Si el usuario lo especifica en la rutina de atención de la interrupción, salva en el stack el contenido de los

registros.

g).- Se atiende a la interrupción, es decir, se realizan las instrucciones pertinentes para satisfacer la solicitud de la interrupción. Al final de esta rutina se rescatan del stack el contenido de los registros, y se habilitan las interrupciones.

h).- Se continúa normalmente con la ejecución del programa. Si se presenta otra interrupción es atendida de la misma manera.

* Referencia 3.5 Y 3.10

3.4.- Software Utilizado.

3.4.1.- Sistema Operativo

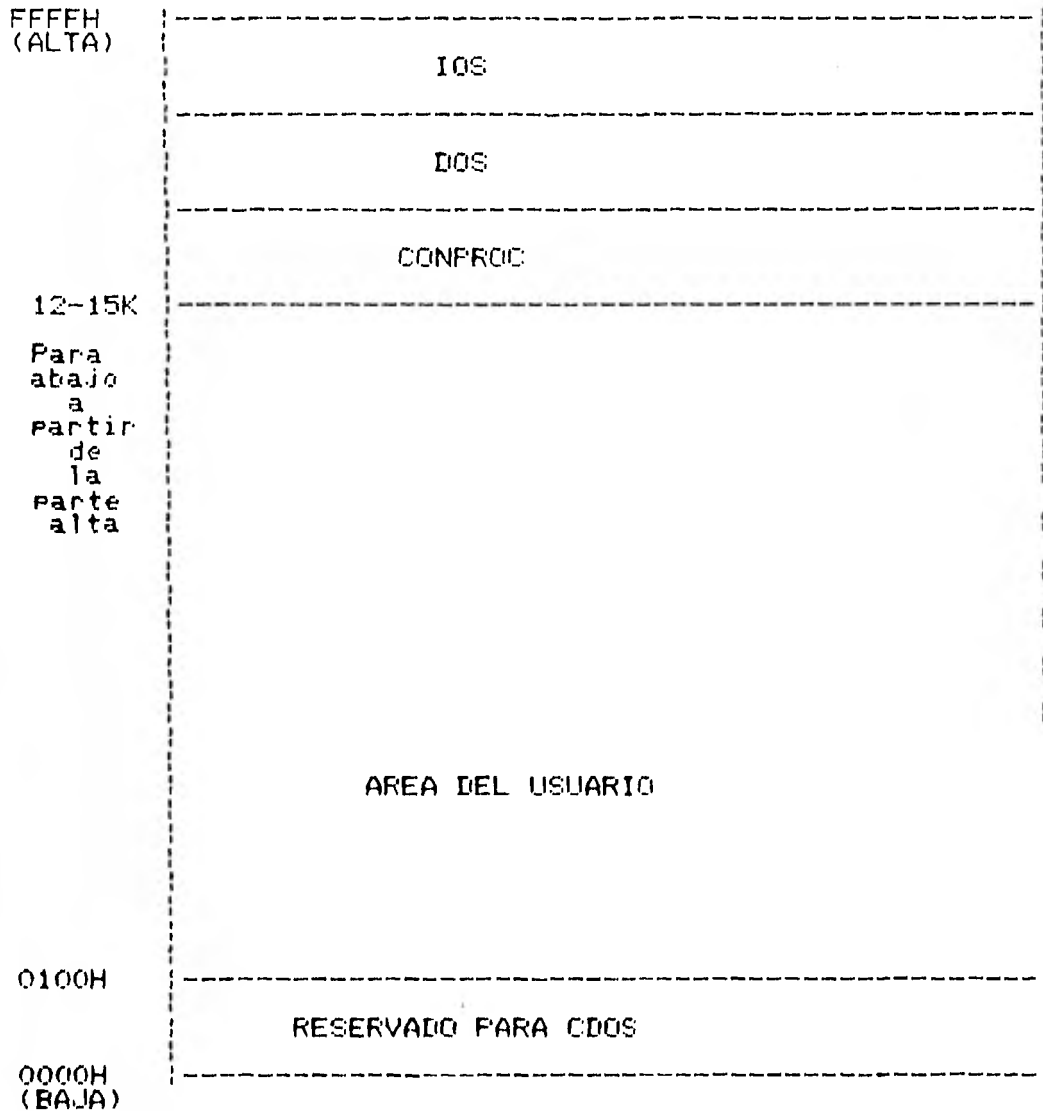
El Sistema Operativo de Disco Cromemco (CDDS) es un producto diseñado y escrito originalmente en código de máquina de Z80 por la compañía Cromemco, para su propia línea de microcomputadoras. Sin embargo debido al gran número de programas que actualmente se pueden conseguir para correr bajo el sistema operativo CP/M, CDDS fue diseñado para ser compatible con CP/M. Cromemco tiene licencia de Digital Data Research (ver apéndice I) para su utilización. Esto significa que la mayoría de los programas escritos para CP/M (de la versión 1.33 en adelante) correrán sin modificaciones bajo CDDS. Esto también significa que los programas escritos para CDDS no siempre funcionarán bajo CP/M. El uso principal de CDDS es el control de Entrada/Salida a los dispositivos de almacenamiento masivo, tales como el disco duro o el floppy. Está diseñado para que los usuarios de las microcomputadoras puedan generar y manipular archivos del tipo secuencial y aleatorio utilizando nombres simbólicos.

CDDS contiene además las rutinas de manejo de la impresora, así como un gran número de llamadas para el manejo de pantalla, por medio de lo cual se puede hacer gráficas, manejar una línea de mensaje en la pantalla, etc,etc.

Localización en memoria.

La memoria de la microcomputadora bajo CDOS está dividida en dos grandes partes. La primera parte es una parte de memoria RAM que CDOS se reserva para el mismo. CDOS ocupa la memoria de la localidad 0000H a la 0100H (parte baja) así también se apropia aproximadamente de la parte alta de 11K a 18K de memoria RAM. La segunda parte es la memoria RAM del usuario. La memoria del usuario ocupa la memoria de la localidad 0100H a la parte baja de CDOS. La longitud del área de memoria depende del sistema operativo que se esté utilizando, dependiendo por ejemplo del número de drives que se tengan declarados en el sistema operativo, si se utilizan las teclas de funciones para la terminal, etc,etc. Generalmente esta área es de 48K.

Se muestra un mapa de la memoria con el sistema operativo.



Mapa de memoria para CDOS

Refiriéndonos al mapa de memoria, podemos ver que se encuentra dividida en dos grandes partes, que son las siguientes:

MEMORIA ALTA.

CDOS tiene las rutinas básicas de Entrada/Salida para la consola, impresora, perforadora y lectora de tarjetas, así también las rutinas de acceso al disco. CDOS tiene las funciones de administración de los archivos, que son las responsables de crear, abrir, cerrar, borrar, consultar, leer y escribir los archivos en el disco. Así también es el encargado de llamar a los programas del usuario y editar los comandos de la consola. Y posee algunas funciones internas llamadas comandos intrínsecos.

MEMORIA DEL USUARIO.

Es en esta parte donde los programas corren.

MEMORIA BAJA.

Esta parte de la memoria esta reservada por CDOS para las siguientes funciones:

0000H - 0002H	Vector del sistema Warm Start
0003H	Byte de E/S
0005H - 0007H	Vector de llamadas al sistema para preguntas del usuario
0008H	Especifica si se está corriendo bajo CDOS si se tiene FFH y bajo CROMIX si se tiene CGH
0030H - 0032H	Puntos de ruptura para DEBUG
0038H - 003AH	Salto al mensaje de: "INVALID JUMP"
0040H - 005BH	Reservada para el sistema
005CH - 007BH	File Control Bloks estándares para el usuario
0080H - 00FFH	Buffer estándar de Entrada/Salida al disco.

CDOS es cargado al área del sistema del disco por medio de la rutina de BOOTSTRAP.

* referencia 6.8

3.4.2.- Ensamblador.

El propósito de un ensamblador es el de tener un medio de traducir de nemónicos, fácilmente recordables y entendibles, a código objeto, el cual puede ser cargado en la memoria y ejecutado como un programa. El Macro Ensamblador Relocalizable que reside en el disco para el Z80, es un ensamblador de dos pasadas, que lee el programa fuente de un archivo en el disco, lo ensambla, y produce un código objeto relocalizable y/o un archivo imprimible.

El Lisador/Cargador puede ser utilizado después, para localizar el código objeto en cualquier parte de memoria. El código de máquina completamente ensamblado y lisado debe de ser guardado en el disco como un archivo tipo *.COM para su ejecución como un programa de comandos.

La utilización de un ensamblador relocalizable y un lisador, provee una de las maneras más versátiles de generar programas en lenguaje de máquina para las computadoras en general. Por medio de estos dos programas, es posible el crear y ensamblar un gran número de módulos separados, y posteriormente lisarlos todos juntos al momento de ejecutarlos. O bien se puede lisar un programa previamente ensamblado desarrollado por el usuario a la biblioteca de programas objeto. Además es posible ensamblar programas utilizando un compilador (por ejemplo el FORTRAN), y lisar estos módulos objeto, para generar subrutinas o programas. Al mismo tiempo es posible que el programa se pueda poner y ejecutar en cualquier parte de la memoria. Este ensamblador soporta Macros, así como los condicionales.

* referencia 3.9

3.5.- Definición del problema.

Para todas y cada unas de las aplicaciones de las microcomputadoras mencionadas en incisos anteriores, es necesario enlazar a la Burroughs con la Cromemco, para el intercambio de información. Las necesidades de enlace las podemos dividir en dos grandes grupos:

- i).- Captura de datos
- ii).- Investigación.

Estas necesidades las enunciaremos a continuación:

- i).- Captura de datos.

Se sustituirán las perforadoras de tarjetas convencionales por microcomputadoras; por medio del software adecuado se simulará una perforadora, y el resultado final será un floppy con los datos grabados en él, (ver referencia 3.11 para más información). Este floppy conteniendo las imágenes de tarjetas deberá ser leído por otra microcomputadora, la cual estará enlazada a Burroughs para la transferencia de las "imágenes" de tarjetas, esto será una lectora de floppies. Es en este tema donde se encuentra enmarcada la presente tesis.

- ii).- Investigación.

Una estación de investigación podrá comunicarse como una terminal remota del computador central; y además tener la capacidad de editar, compilar y ejecutar "localmente" los programas

escritos en lenguajes tales como: Cobol, Basic, Fortran. Así mismo podrá tener tarjetas controladoras de periféricos y de conversión Analógico/Digital para tomar muestras de señales externas y el resultado grabarlo en disco, para posteriormente procesarlas localmente, o bien transferirlas al computador central.

En la actualidad, las terminales de investigadores, solamente realizan un papel estático, pues su única función consiste en terminales remotas de programación. Con un sistema de este tipo se contará con dos computadoras en una. Es posible el editar, compilar, adquirir datos y programas en forma local, y también contar con una terminal de Burroughs para transferir los programas editados, recibir los resultados de dichos programas y además imprimir los resultados localmente si así se desea.

Por lo dicho anteriormente, es necesario desarrollar el "software" necesario para manejar un puerto serie a través del cual se conectará el computador central y simular una terminal remota. Así mismo poder leer de disco un archivo determinado y transmitirlo al computador central. El recibir un archivo de Burroughs, almacenarlo temporalmente en la memoria y posteriormente grabarlo en disco. Con estas tres rutinas básicas se cubren casi todas las necesidades del enlace de una microcomputadora con la Burroughs.

Existen en el mercado diversos programas para la comunicación de una microcomputadora con una macrocomputadora, pero todos estos tienen algunas limitaciones en cuanto a capacidad o manera de conexión. Se decidió desarrollar este sistema por las necesidades antes expuestas de que cualquier usuario sin experiencia previa en

microcomputadoras pudiera utilizarlas, puesto que el programa será auto-explicativo y además será en Español. Con lo que perforistas, operadores y estudiantes en general, podrán utilizarlo.

Por otra parte, es necesario tener el programa fuente para poder realizar los cambios necesarios.

Es necesario desarrollar el software para una lectora de discos, es decir un programa que pida el nombre del archivo que será transmitido, investigar si existe, si así es, leer el archivo y transmitirlo al computador central. De preferencia el mismo programa de transmisión y recepción de archivos podrá hacer esto, solo es necesario realizar algunos parches para que el sistema funcione correctamente.

Inicialmente es necesario realizar un programa no muy complicado, pero lo suficientemente inteligente para suplir todas las necesidades expuestas anteriormente.

Después se puede desarrollar una versión más compleja, pero se puede aprovechar la experiencia adquirida con el primero.

3.5.1.- Características del enlace.

Para la realización de este sistema, se deberá contar con un puerto serie capaz de trabajar a diferentes velocidades, para esto se seleccionó a la tarjeta de hardware "TU-ART" (ver inciso 3.3.2). Esta tarjeta cuando recibe un carácter, puede generar una petición de interrupción (si así se desea), para que el procesador reciba el carácter. Para trabajar por interrupciones es necesario programar los puertos para indicarle cuando interrumpa, cual será la dirección que nos proporcione para la atención de la interrupción. Los dispositivos que nos pueden interrumpir en el enlace con la Burroughs, es el teclado de la consola del micro y la línea del computador central.

Otra forma de resolver esto, es estar en un loop presuntando continuamente el estado del micro y de la línea del computador central. Cuando se presenta un carácter en alguno de los dos dispositivos, se lee el carácter y se realizan con él las operaciones pertinentes.

El programa de comunicación con la Burroughs se realiza utilizando la técnica de "polling" descrita anteriormente.

El diagrama de bloques de las dos máquinas para el enlace es el siguiente:

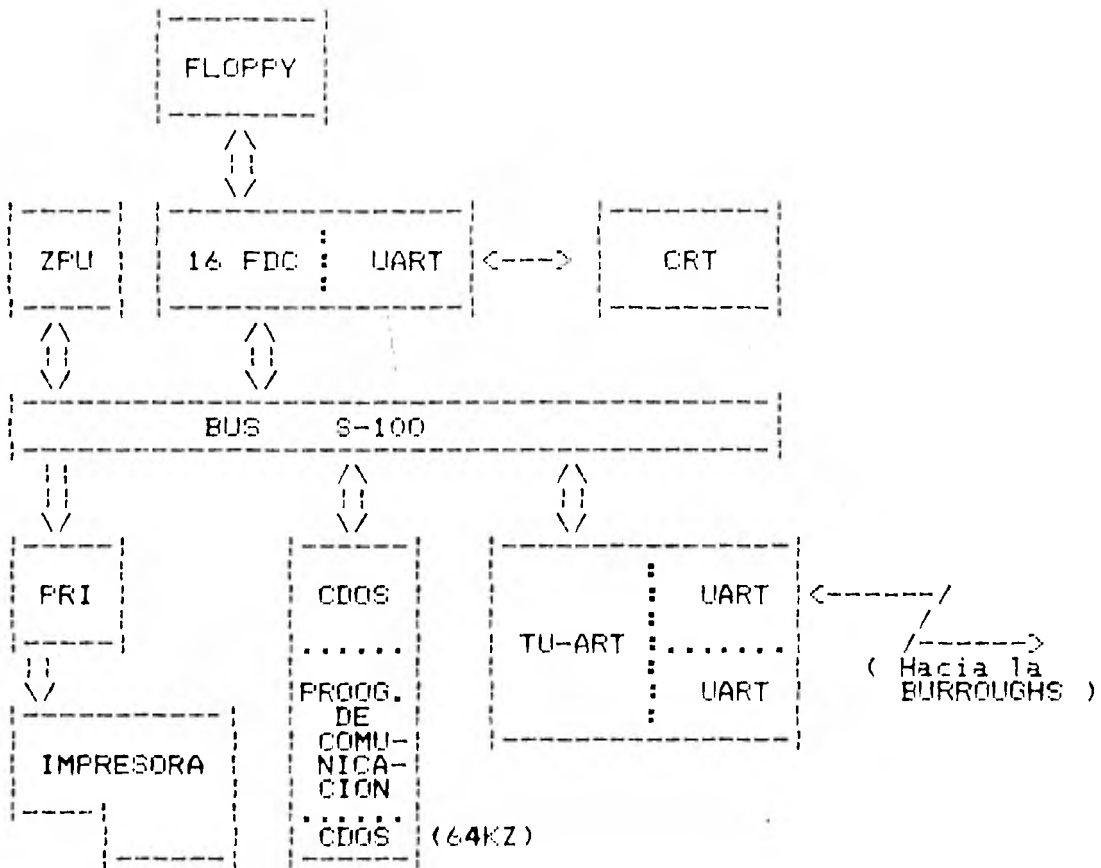


Diagrama del sistema y enlace con el Computador Central.

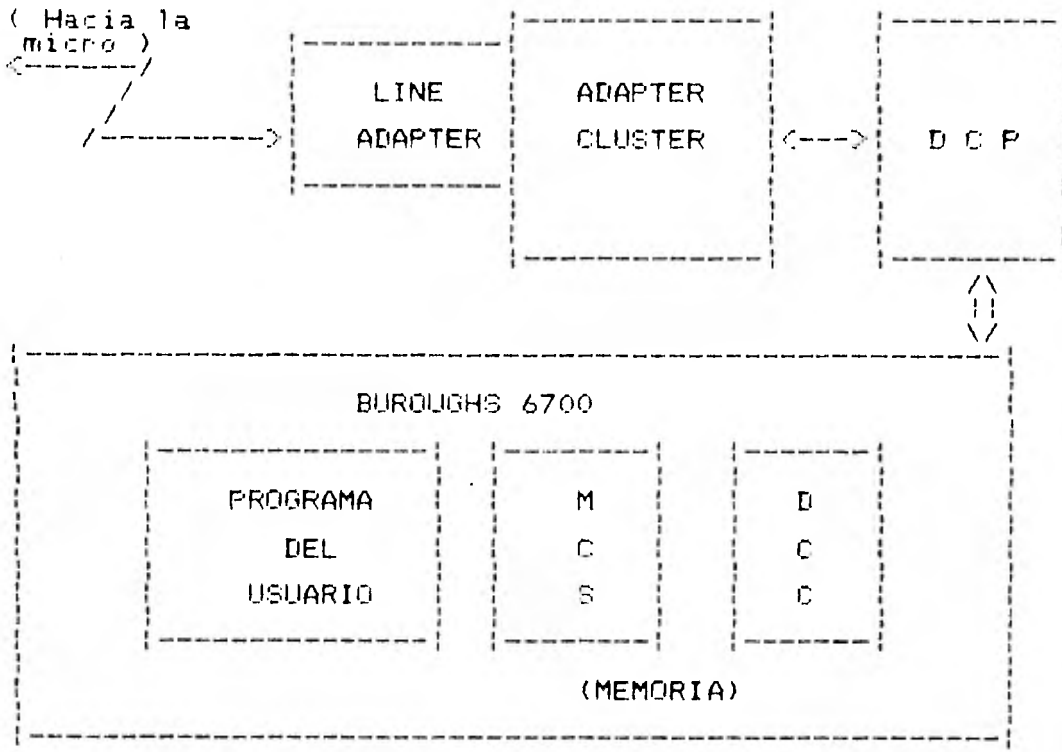


Diagrama de bloques del Computador Central.

Para la utilización eficiente de la interfaz de comunicación asíncrona TU-ART, el programa se escribirá en Ensamblador, (ver inciso 3.4.2) por medio del cual se tiene un control completo de los puertos de esta tarjeta, y del puerto de la consola del operador, así mismo son posibles los accesos a disco, para la parte de recepción y transmisión de archivos.

El sistema deberá ser capaz de operar a varias velocidades, pues deberá soportar la conexión a través de Módem, Line-Driver o directamente. Deberá contar también con el Break.

Cuando un usuario se encuentre utilizando la microcomputadora como una terminal de Burroughs, este no deberá notar que se encuentra conectado a través de una microcomputadora, pues todas las teclas que tiene una terminal remota normal tendrán la misma función. Con esto el usuario puede tener acceso a Cande, y realizar todas las funciones de que éste dispone.

El enlace se realiza en forma asíncrona y por un canal de comunicación tipo serie, se hace de esta manera, pues estas son las características más comunes de las terminales de teleproceso con que cuenta la Universidad.

3.5.2.- Limitaciones y posibles mejoras.

En la utilización del sistema para la comunicación de la microcomputadora Cromemco con el computador central Burroughs, el usuario que esté trabajando a través del primero como una terminal del segundo, verificará en línea que lo recibido y lo transmitido, sea lo que se desea. Si la línea de comunicación entre ambas máquinas tiene mucho ruido, el usuario puede detectarlo fácilmente, pues la información será errónea. Este sistema no tiene manera de detectar los errores y menos aún de corregirlos; es esta su principal desventaja.

En la rutina de transmisión de archivos, se tienen limitaciones mucho más serias, estas son:

- Cuando se está transmitiendo, no es posible saber si todo lo transmitido está llegando correctamente al computador central. En algún momento de la transmisión, la línea de intercomunicación puede estar fuera de servicio y la microcomputadora no se dará cuenta de esto. Esto sucede, puesto que no se tiene ningún protocolo de comunicación que verifique que se reciba correctamente la información.
- Si la energía eléctrica se suspende temporalmente, al reanudarse el servicio, no se puede transmitir a partir del registro donde se encontraba, pues se perdió toda la información que teníamos en la memoria, por eso es necesario iniciar nuevamente la transmisión.

La principal desventaja está en la recepción de archivos, pues el tamaño máximo del archivo que se puede recibir es el del tamaño de la memoria, menos la longitud del sistema operativo, menos la longitud del programa; es decir:

64K	Memoria total
- 10K	Longitud del Sistema Operativo
- 4K	Longitud del Programa de Comunicación

50K	Longitud de la memoria disponible para la recepción del archivo.

Se reciben los 50K del archivo en la memoria; pues no es posible realizar accesos a disco, mientras estos se realicen se perderían datos; y cuando la memoria está llena, se suspende la recepción y se graba en el floppy lo recibido. Si la energía eléctrica se suspende en este momento, lo que no se encuentra grabado en el disco, en este momento se borra y no se puede recuperar la información que teníamos en la memoria.

Otra posible solución, es la de establecer un pequeño "diálogo" entre ambas computadoras; este "diálogo" será utilizado solamente en la parte de transmisión y recepción.

El "diálogo" consiste en los siguientes puntos:

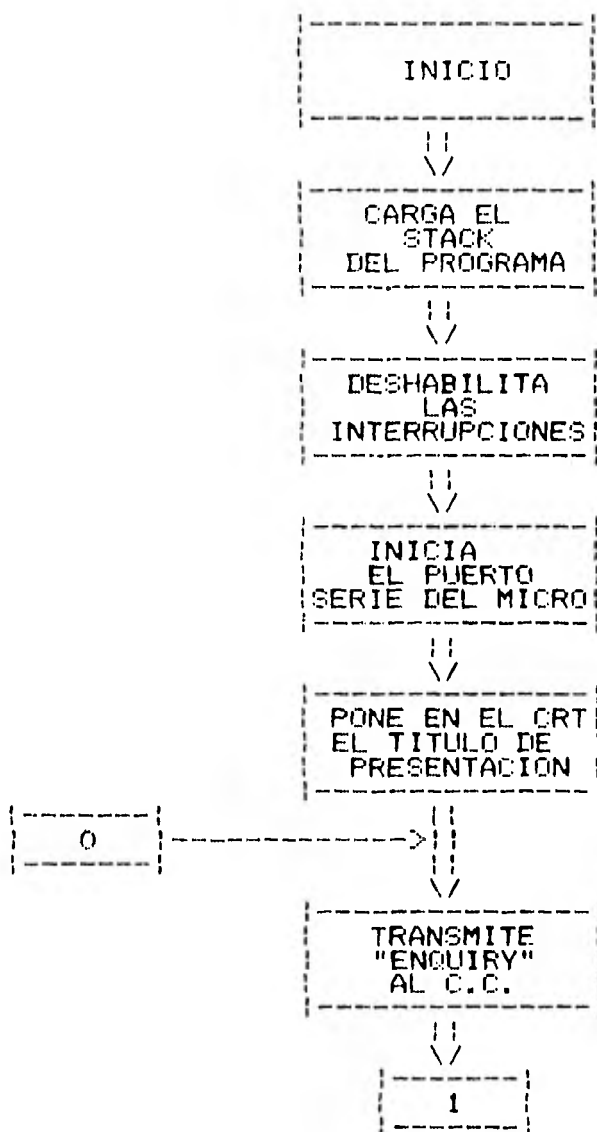
- a).- El computador central transmite un registro, y espera una contestación.
- b).- La contestación puede ser X-ON ó bien X-OFF.
- c).- En caso de ser X-ON, pasa al inciso (a).
- d).- Si la contestación es X-OFF, esperará hasta recibir un X-ON, y pasará al inciso (a).
- e).- La microcomputadora al recibir el registro, lo guarda en la memoria, e investiga si aún tiene espacio disponible en la memoria.

- f).- Si el espacio disponible en la memoria es suficiente, transmite un X-ON al computador central y espera otro registro.
- g).- En el caso de que no tenga espacio, transmitirá un X-OFF al computador central para que este deje de enviarle y pueda guardar todo lo recibido en el disco. Cuando haya terminado de guardarlo, transmite X-ON al computador central para que continúe con la transmisión.

Este sistema se realiza como una primera versión del enlace de la Cromemco con la Burroughs. Esto se hace con el fin de que la experiencia adquirida, posteriormente se utilice para un sistema con protocolo y posiblemente que varios usuarios compartan el puerto de comunicación, si así se desea, o bien que trabajen algunos en forma local y otros en enlace con el computador central. Con esto se cumplen los requisitos establecidos en las secciones anteriores. En los apéndices se hablará más del enlace con protocolo.

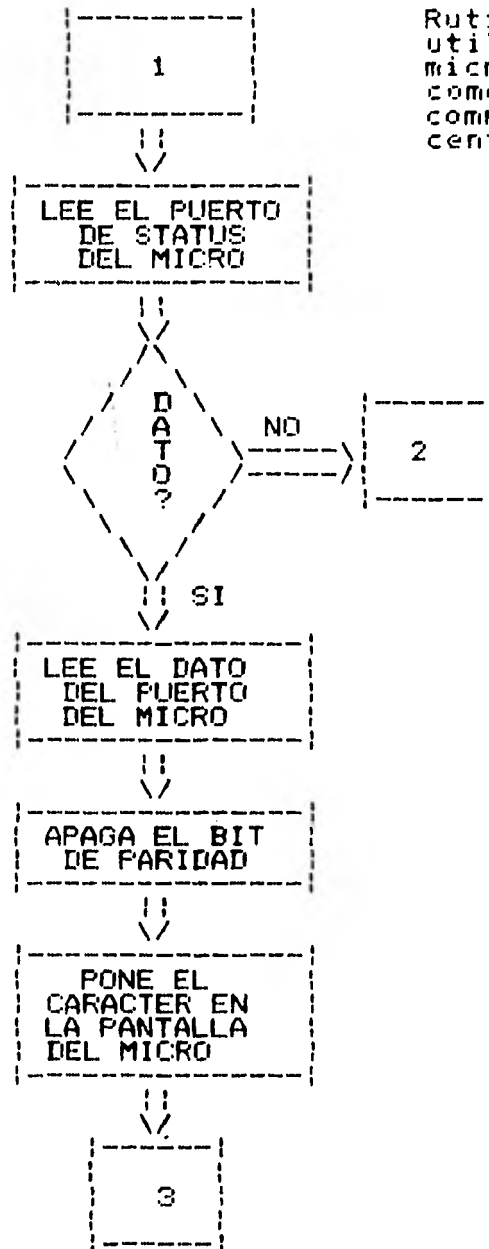
3.5.3.- Diagrama de flujo.

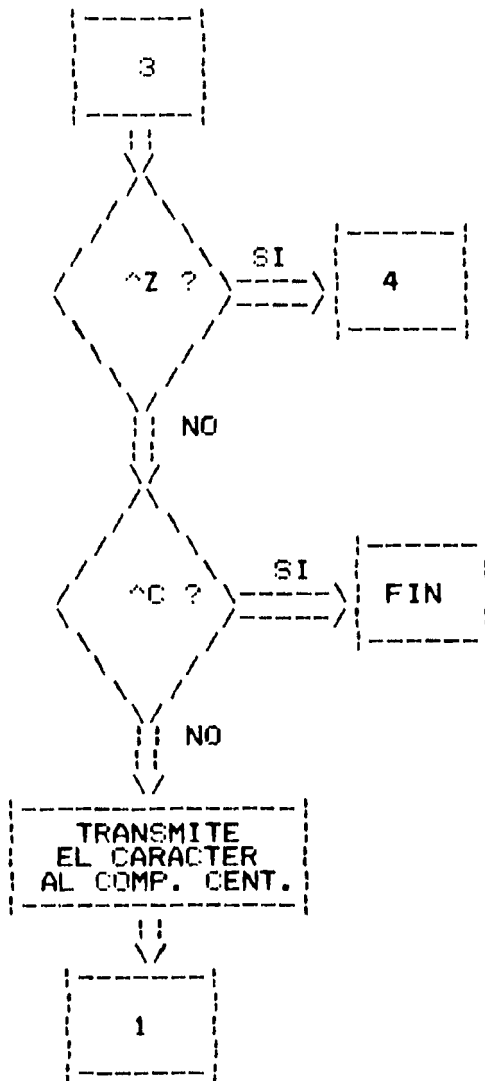
A continuación se presenta el diagrama de flujo del sistema de transferencia de archivos. Se presenta como diagrama de bloques, pues si se desea más información, con leer la parte del programa deseada es suficiente, pues está ampliamente explicada.

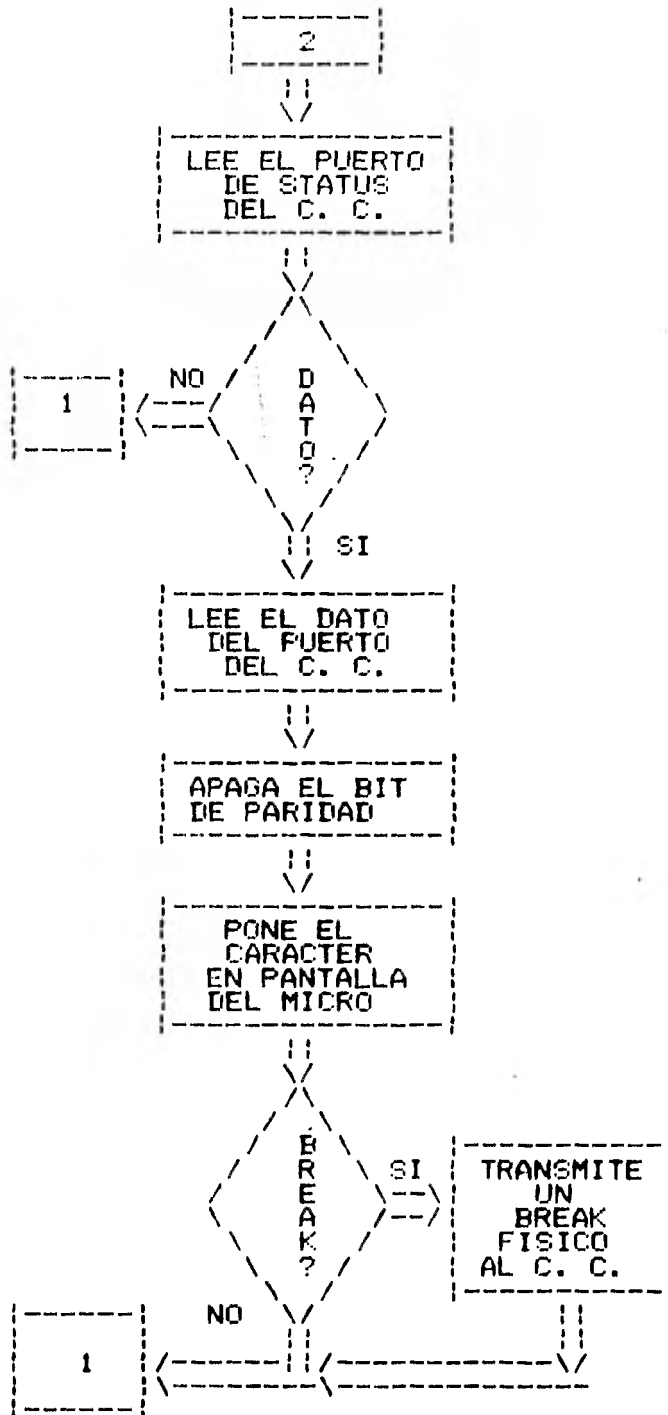


Rutina para el inicio de los puertos, así como de las variables que se utilizan durante su ejecución.

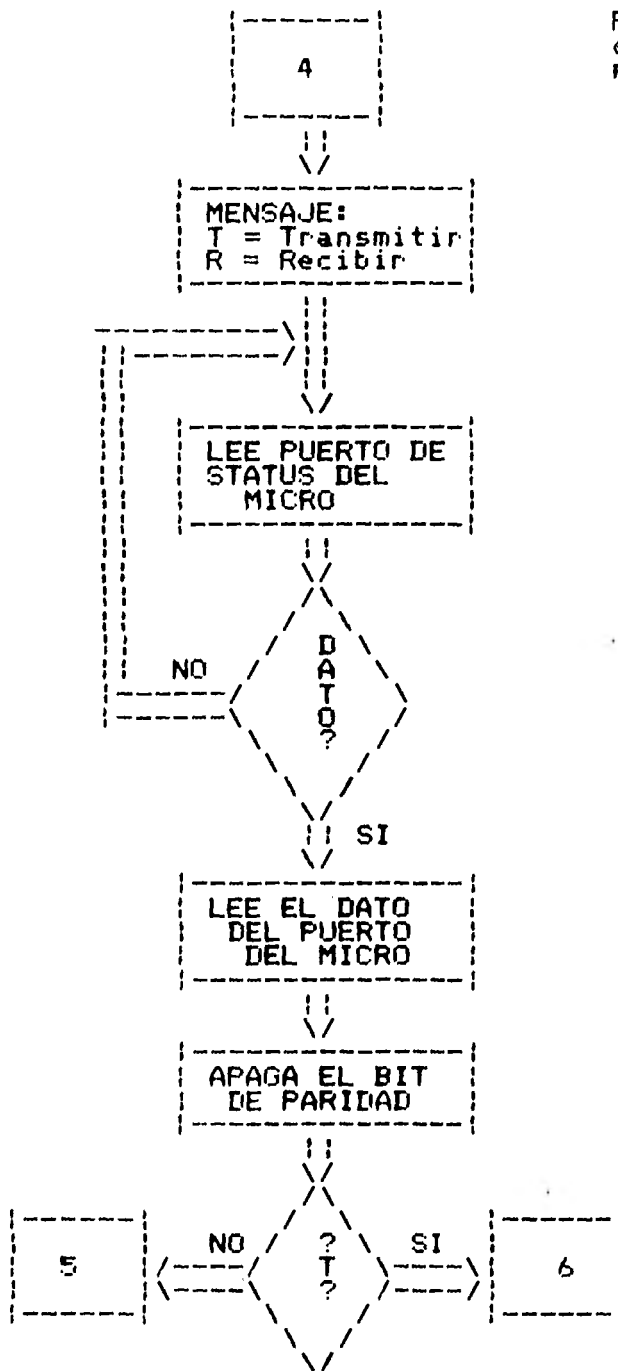
Rutina para la
utilización del
microprocesador
como terminal
computador
central.

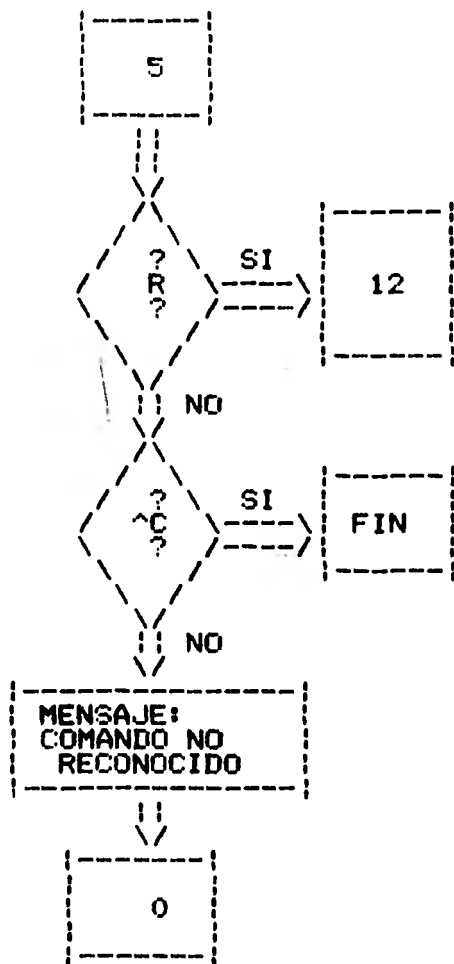




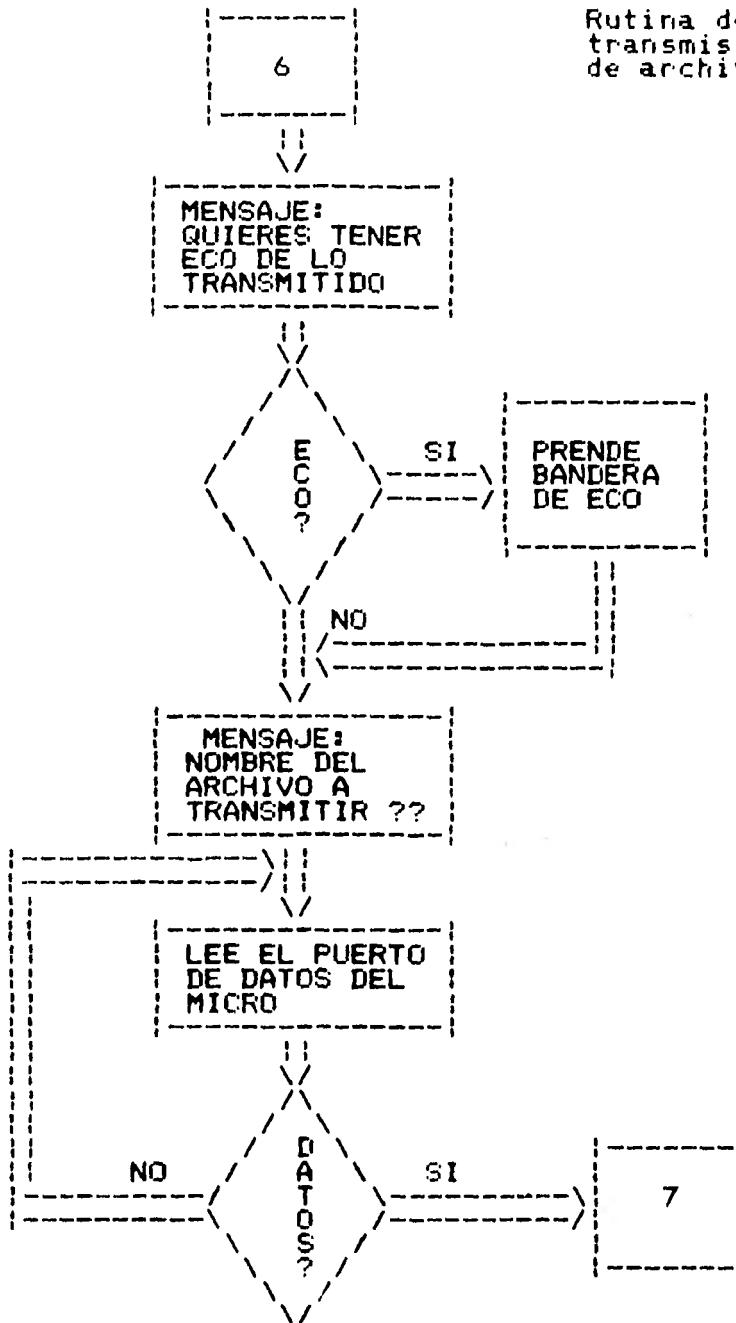


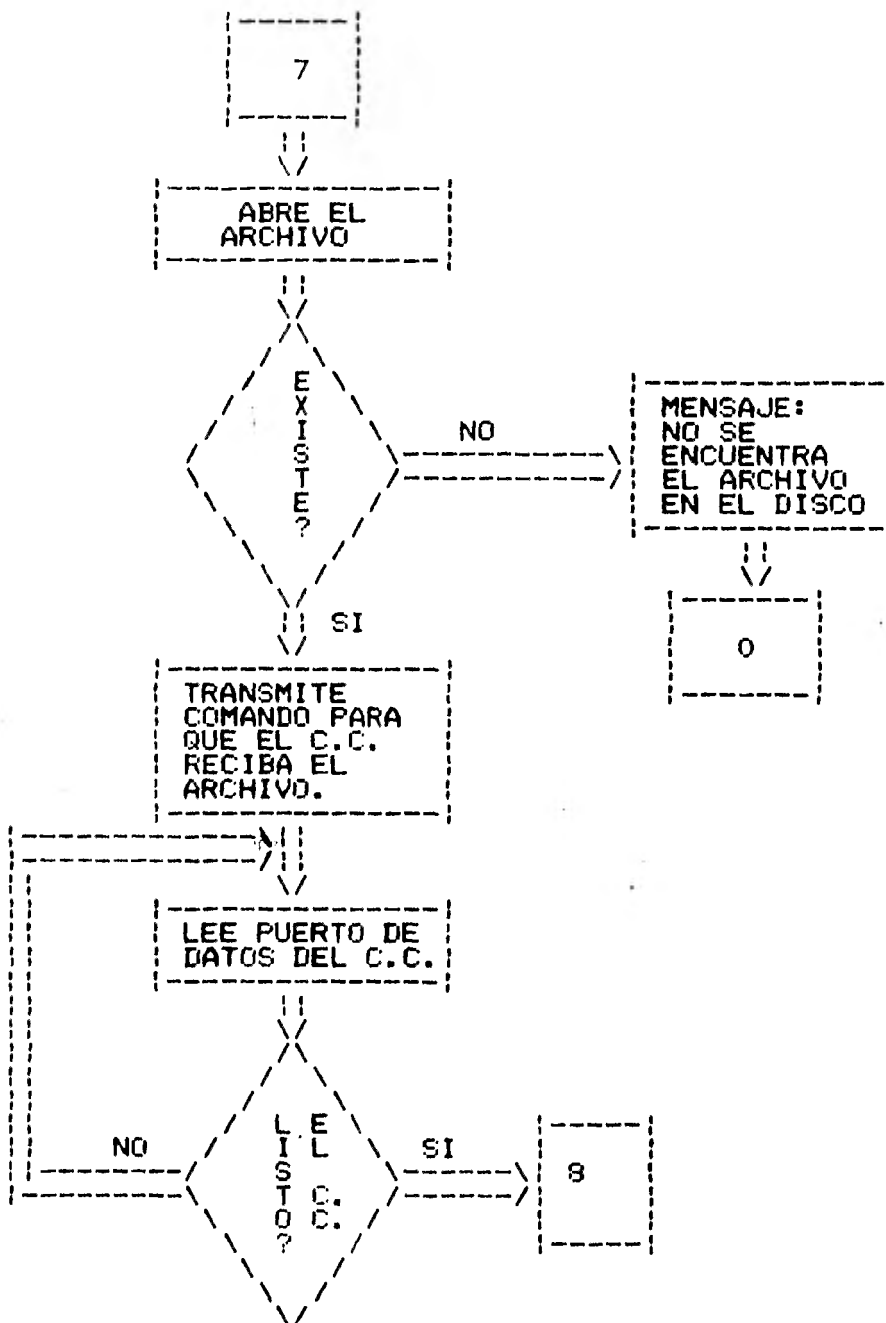
Rutina para
el menú del
Programa.

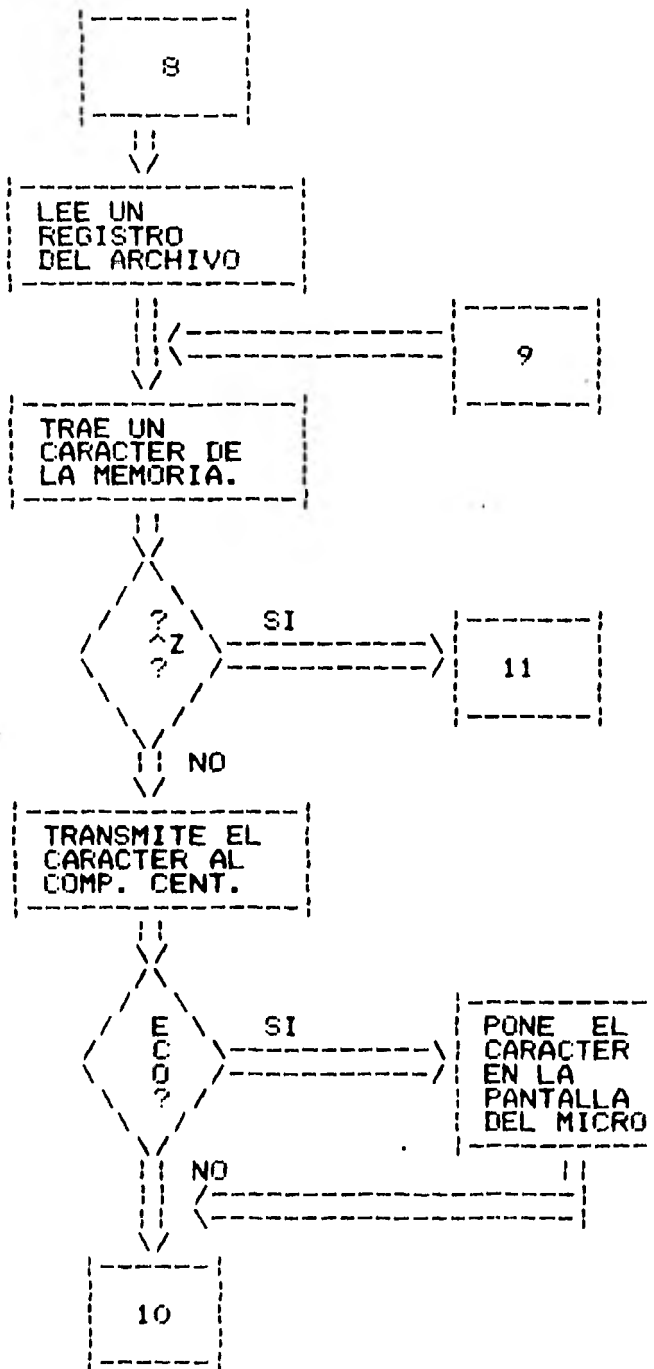


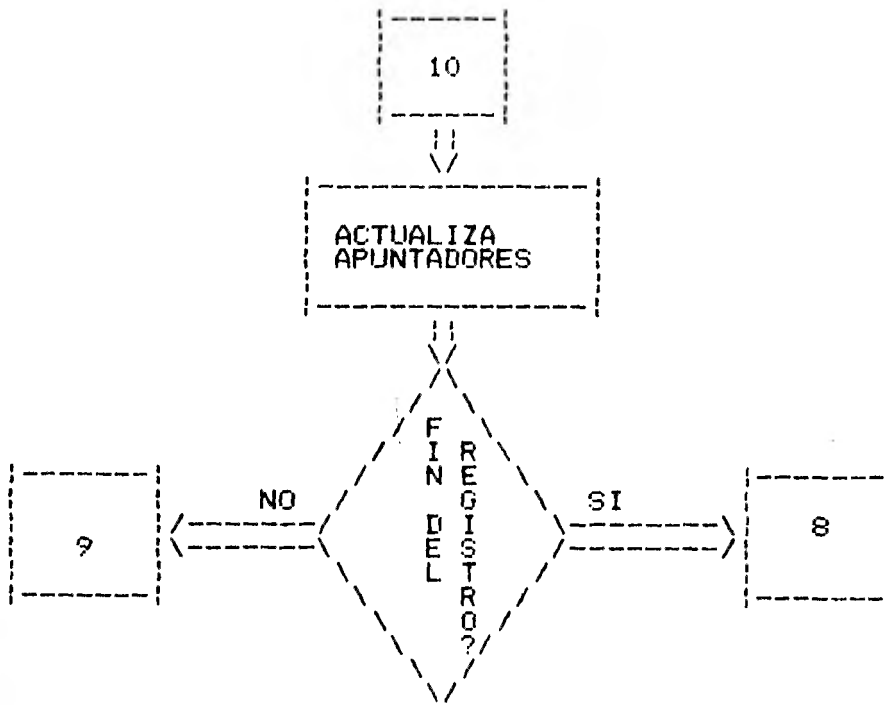


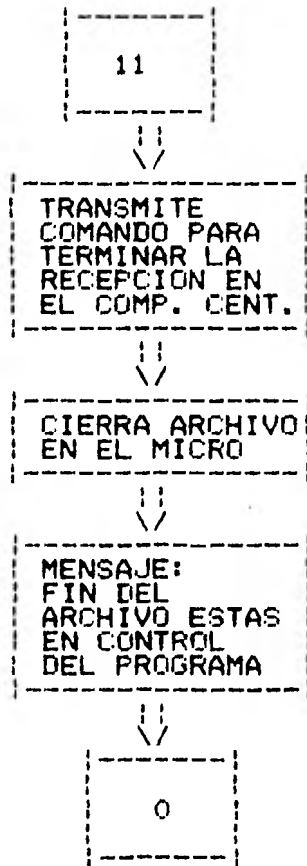
Rutina de
transmisión
de archivos.



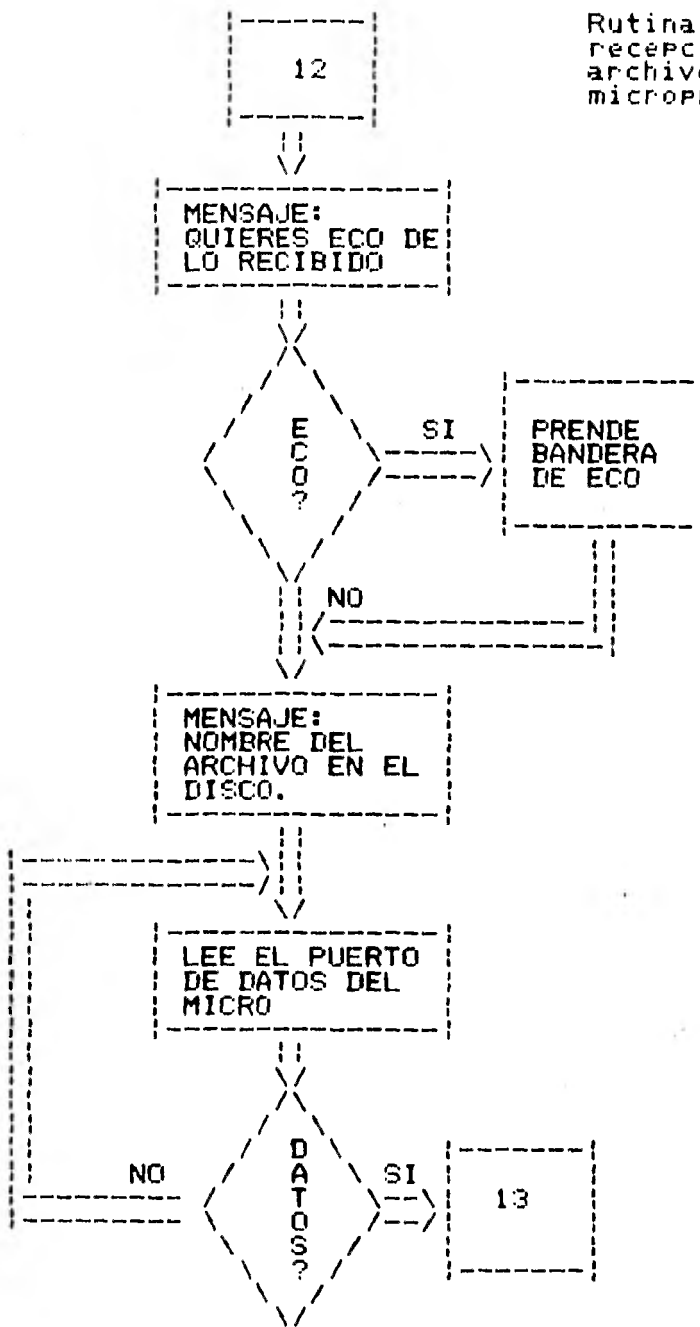


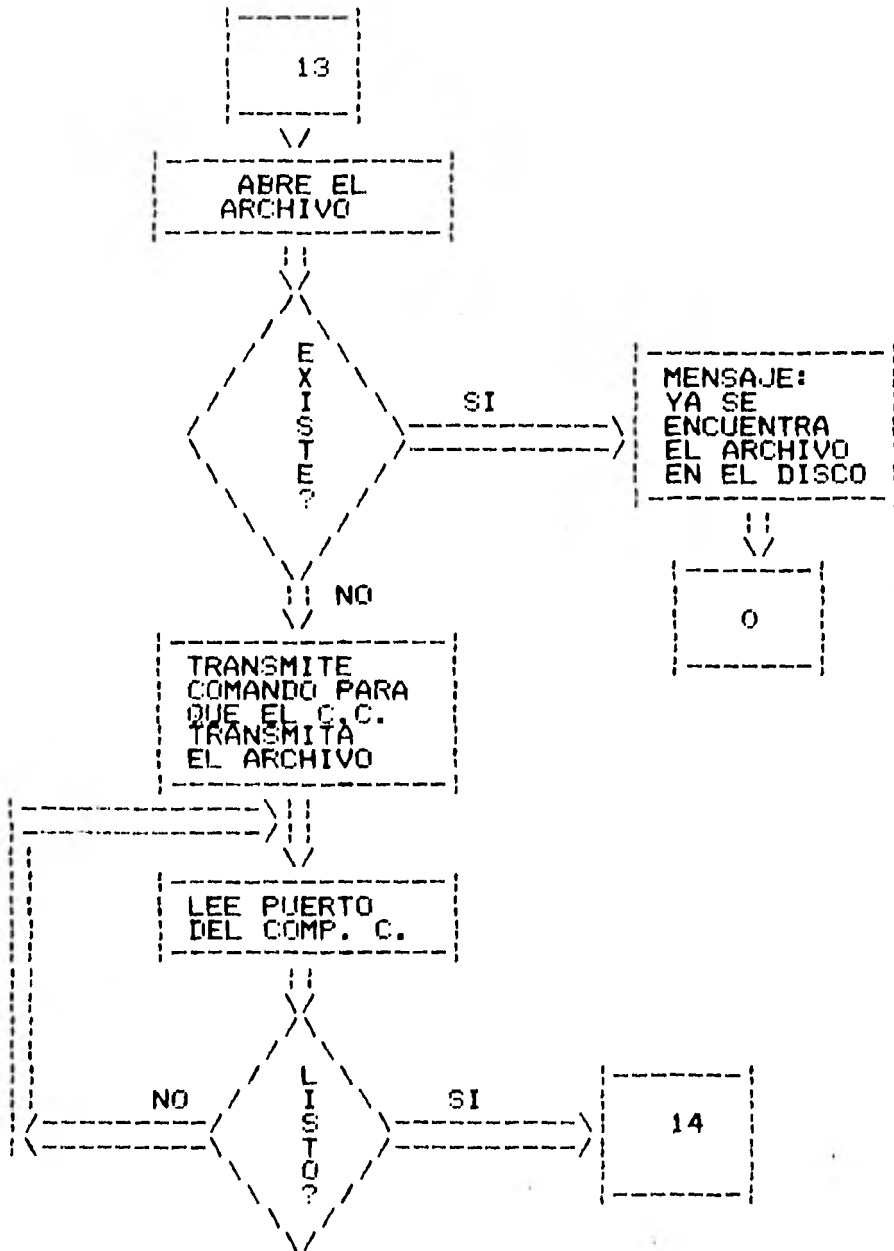


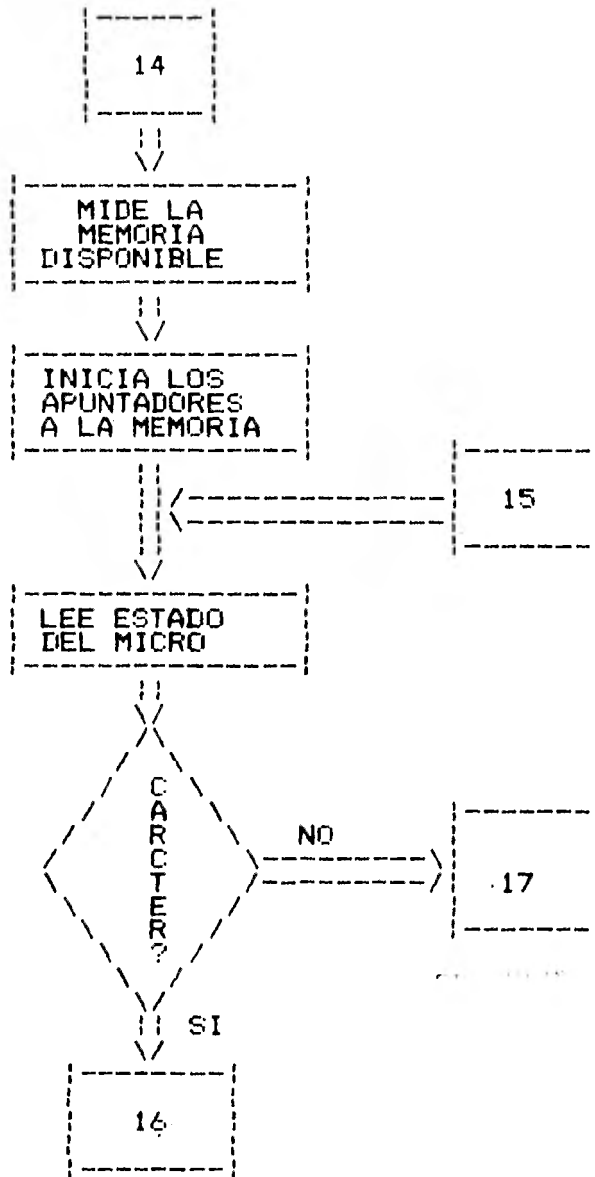


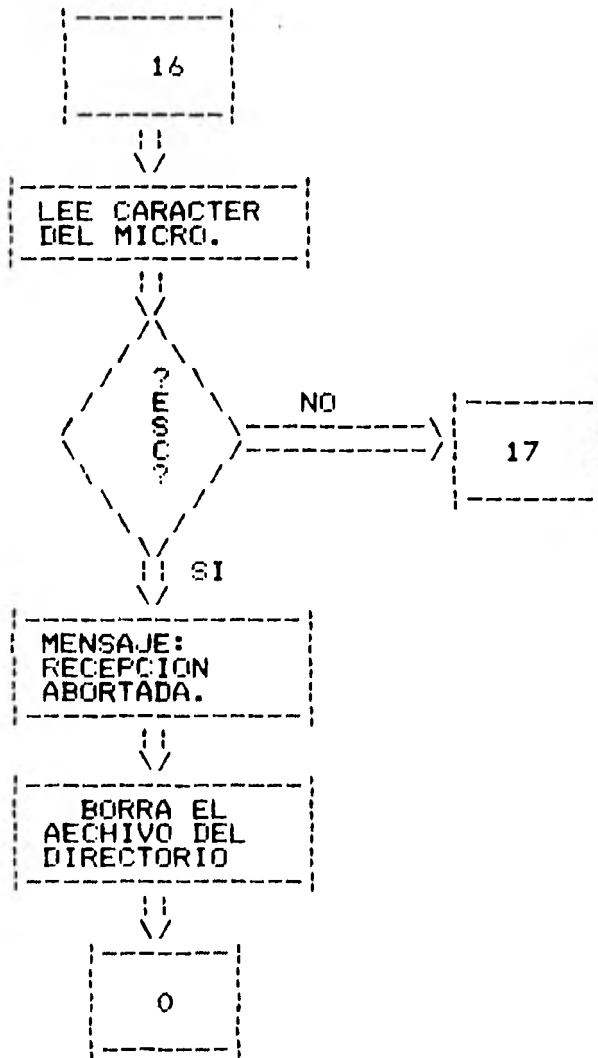


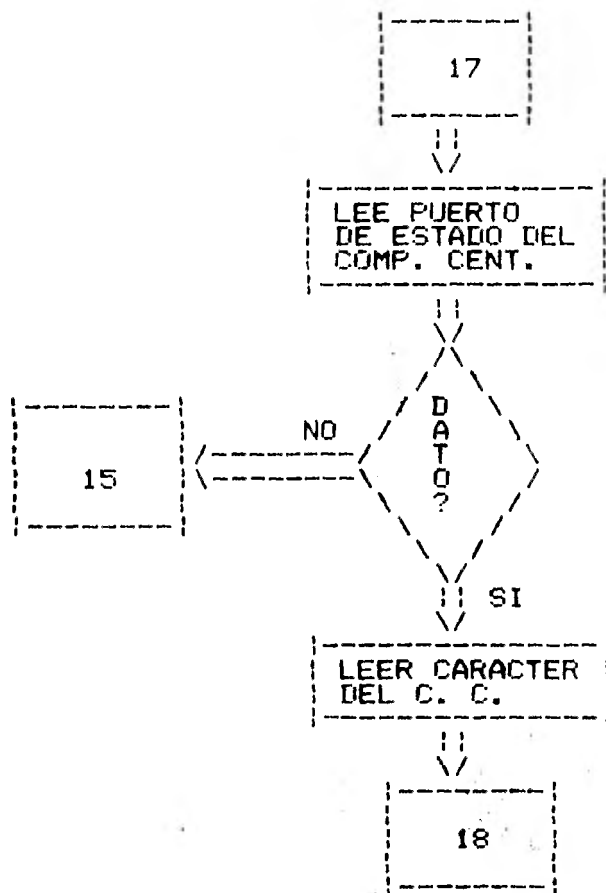
Rutina para la
recepción de
archivos en el
microprocesador.

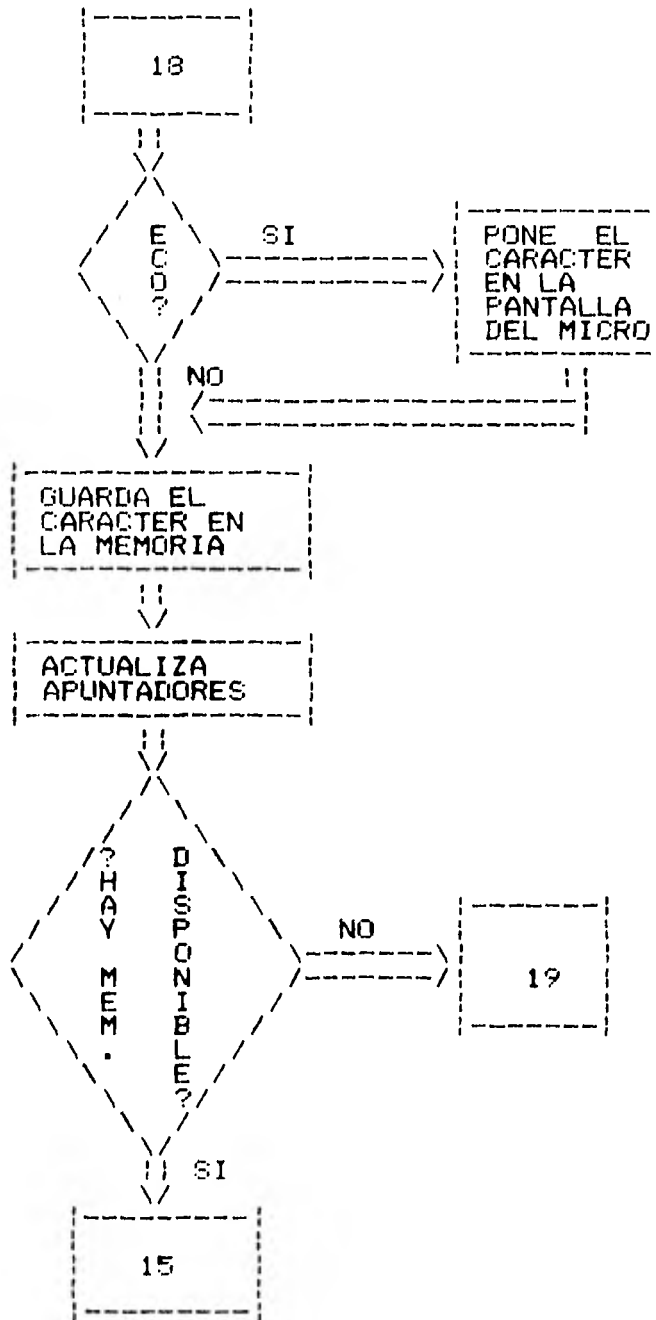


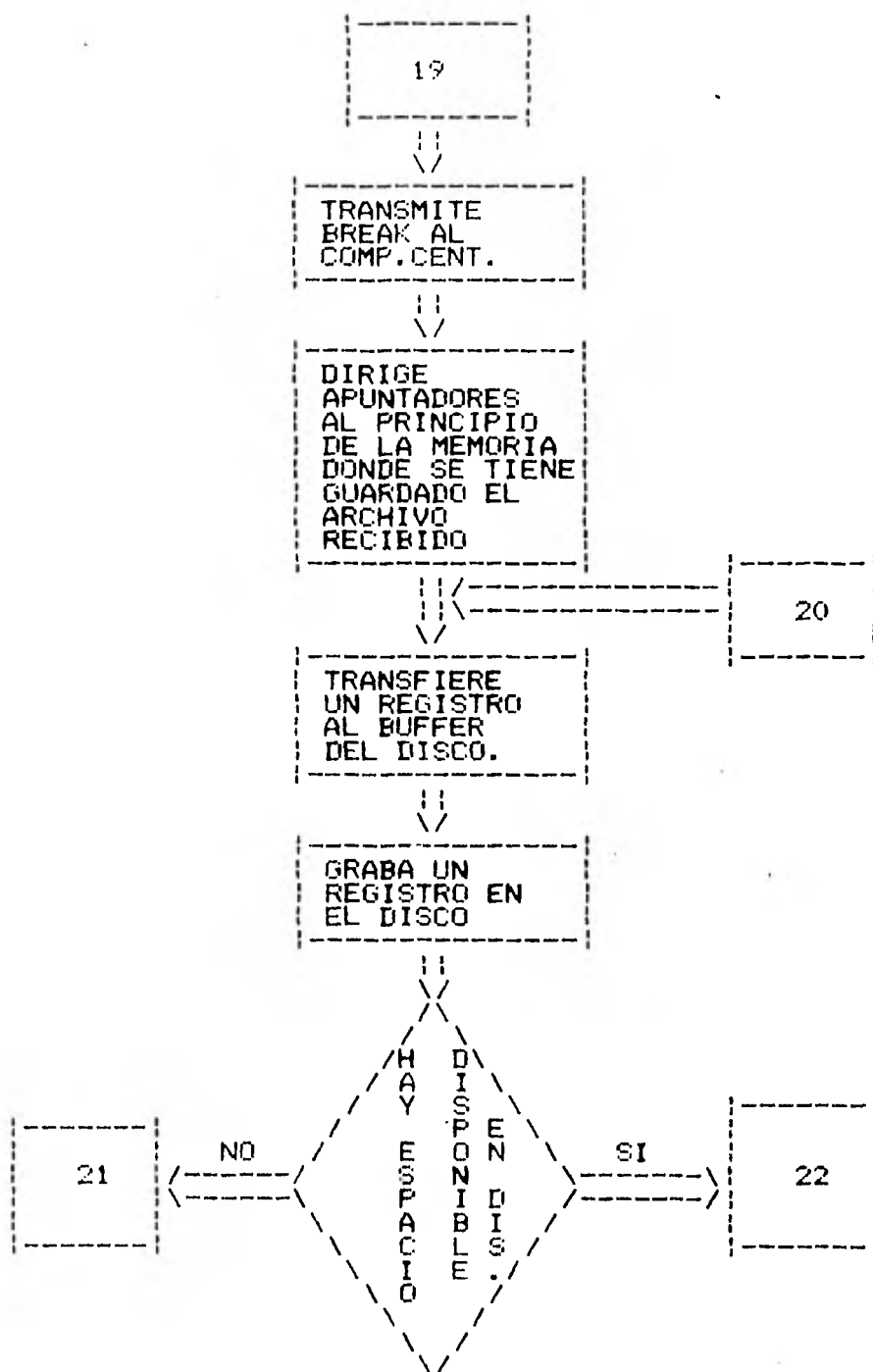


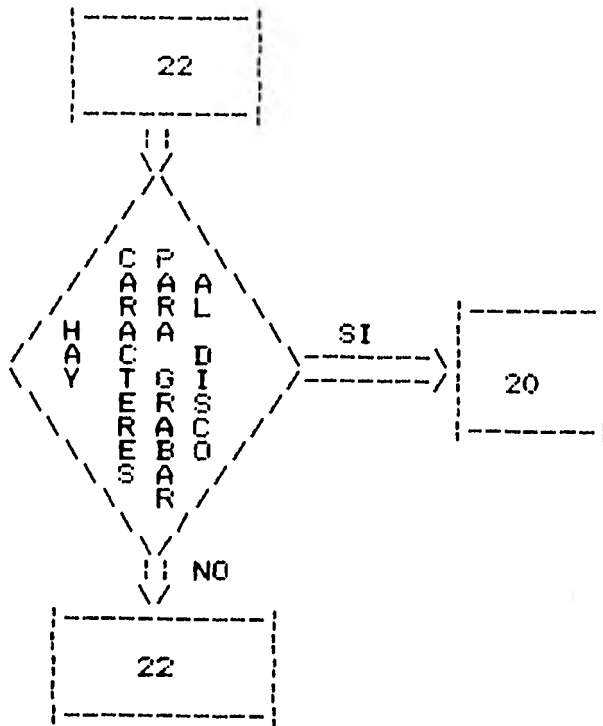


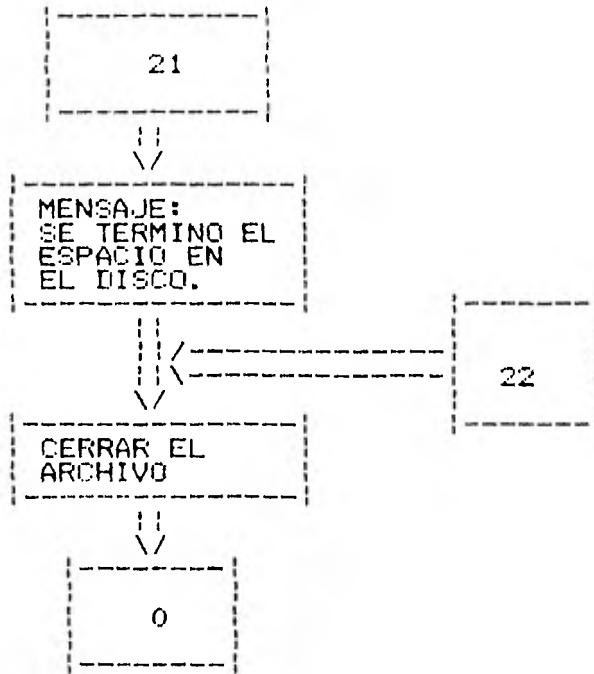












3.5.4.- Listado del programa.

A continuación se incluye un listado del programa de comunicación de una microcomputadora CROMEMCO con microprocesador Z80 y sistema operativo CP/M.

```
0008 ;
0009 ;
0010 ; Este programa permite enlazar una macrocomputadora con
0011 ; el sistema CROMEMCO.
0012 ;
0013 ; El metodo utilizado es el siguiente:
0014 ; 1) EN MODO LOCAL.
0015 ; La microcomputadora recibe caracteres del computador
0016 ; central o del sistema microprocesador y hace "eco" en el otro
0017 ; sistema. Por ejemplo: Si la computadora central transmite,
0018 ; el microprocesador recibe el caracter y lo pone en la
0019 ; pantalla de el mismo. Si el microprocesador transmite, hace
0020 ; eco en la pantalla del caracter tecleado y lo transmite a la
0021 ; linea del computador central.
0022 ;
0023 ; 2) EN MODO DE TRANSMISION.
0024 ; El sistema microprocesador hace eco (si asi se desea) en
0025 ; la pantalla de todo aquello que se toma del disco del
0026 ; microprocesador y lo transmite al computador central.
0027 ;
0028 ; 3) EN MODO DE RECEPCION.
0029 ; Todo lo que se reciba del computador central es suardado
0030 ; en la memoria local temporalmente, para posteriormente
0031 ; suardarlo definitivamente en el disco local.
0032 ;
0033 ; Notas aclaratorias.
0034 ;
0035 ; Es importante mencionar que al TU-ART (tarjeta de
0036 ; hardware donde se conecta fisicamente el computador central)
0037 ; debe de estar conectada una linea del computador central,
0038 ; para que el sistema pueda funcionar, de lo contrario, NO
0039 ; funcionara.
0040 ;
0041 ; Asi mismo el sistema CROMEMCO debe de estar debidamente
0042 ; aterrizado, pues de otra manera se recibira basura, y el
0043 ; equipo puede danarse.
0044 ;
0045 ; Programa realizado por:
0046 ;
0047 ;
0048 ;
0049 ;
0050 ; Mario Alberto Rivera Ponce.
0051 ;
0052 ;
0053 ;
0054 ;
0055 ;
0056 ;
```

CROMEMCO Z80 Macro Assembler Version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de variables para el ensamblado.

Sep 07, 1982 20:37:04

Page 0002

```
0058 ;
0059 ;
(FFFF) 0060 TRUE EQU -1
(0000) 0061 FALSE EQU 0
0062 ;
0063 ;
0064 ;
0065 ; UNA Y SOLO UNA DE LAS SIGUIENTES VELOCIDADES DEBERAN SER VERDADERAS.
0066 ;
0067 ;
(0000) 0068 V1200: EQU FALSE ; TRUE cuando esta sea la velocidad
0069
(0000) 0070 V2400: EQU FALSE ; TRUE cuando esta sea la velocidad
0071
(FFFF) 0072 V4800: EQU TRUE ; TRUE cuando esta sea la velocidad
0073
(0000) 0074 V9600: EQU FALSE ; TRUE cuando esta sea la velocidad
0075
(FFFF) 0076 C3102: EQU TRUE ; TRUE cuando se utilice la terminal
0077 ; CROMEMCO 3102.
0078
(0000) 0079 LECTORA: EQU FALSE ; TRUE cuando solo se quiera una
0080 ; lectora de floppy.
0081
(0000) 0082 MICROS: EQU FALSE ; Cuando el enlace sea entre micros.
0083
(0000) 0084 DISKS: EQU FALSE ; Cuando se tensan discos de este tipo.
0085
(0000) 0086 MODEM: EQU FALSE ; Cuando el enlace sea a traves de modem.
0087
(FFFF) 0088 SYSZERO: EQU TRUE ; Cuando se utilice este sistema.
0089
0090
0095
0100
(00A0) 0102 VELOCIDAD: EQU 0A0H ; 4800 baud / 1 stop bit
(00FF) 0103 RETARDO: EQU 00FFH ; 0256
0105
0110
0111
0112
```

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Configuracion del TU-ART

Pase 0003

```
0114  
0115 ;  
0116 ;      *** DEFINICION DE CONTROLES DEL TU-ART ***  
0117 ;  
(0009) 0118 RESET: EQU    09H    ; Reset al Puerto de datos del Uart.  
(0002) 0119 BRK:   EQU    02H    ; Break al computador central.  
(0004) 0120 SRV:   EQU    04H    ; Break detectado desde la consola del micro.  
(0040) 0121 RDA:   EQU    40H    ; Mascara para Read data available.  
(0074) 0122 PSAR:  EQU    74H    ; Parallel Output Port.  
(0080) 0123 TBE:   EQU    80H    ; Mascara para Transmit buffer empty.  
0124  
(0000) 0125 CNTMICRO: EQU    00H    ; Puerto de control de  
0126 ;      la consola del micro.  
(0000) 0127 ABASE:  EQU    CNTMICRO ; Puerto base de la consola.  
(0001) 0128 DATMICRO: EQU    CNTMICRO+1 ; Mascara para el Puerto de  
0129 ;      datos del micro.  
0131  
(0050) 0132 CNTLINK: EQU    50H    ; Puerto de control HOST.  
0133  
0134  
0140  
(0050) 0141 PBAUD:  EQU    CNTLINK  ; Puerto de baudaje del C.C.  
(0051) 0142 DATLINK: EQU    CNTLINK+1 ; Puerto de datos del HOST.  
(0052) 0143 PCOMMAND: EQU    CNTLINK+2 ; Puerto de comandos del c.c.  
0144  
0145
```

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de caracteres ASCII.

```
0147
0148 ;*****
0149 ;*
0150 ;*
0151 ;*
0152 ;*
0153 ;*****
0154
(0000) 0155 NULL EQU 00H ; Character ASCII de NULL.
(0003) 0156 CTRLC: EQU 03H ; Character ASCII de CTRL-C.
(0005) 0157 CTRL E: EQU 05H ; Character ASCII de CTRL-E.
(0007) 0158 BEL: EQU 07H ; Character ASCII de BELL.
(0008) 0159 BS: EQU 08H ; Character ASCII de BACK SPACE.
(000D) 0160 CR: EQU 0DH ; Character ASCII de CR.
(000A) 0161 LF: EQU 0AH ; Character ASCII de LF.
(0011) 0162 DC1: EQU 11H ; Character ASCII DE X-ON.
(0012) 0163 DC2: EQU 12H ; Character ASCII de END.
(0013) 0164 CTRLS: EQU 13H ; Character ASCII de CTRL-S.
(0013) 0165 DC3: EQU 13H ; Character ASCII de x-off.
(001A) 0166 CTRLZ: EQU 1AH ; Character ASCII de CTRL-Z.
(001B) 0167 ESC: EQU 1BH ; Character ASCII de ESC.
(0020) 0168 SPACE: EQU 20H ; Character ASCII de SPACE.
(007F) 0169 DEL: EQU 7FH ; Character ASCII de DELETE.
0170
```

C.S.C. Sistema de Transferencia de Archivos Version 04.01.
Definicion de localidades del sistema operativo CDOS, Y Macros.

```
0172
0173
0174 |-----|
0175 |         |
0176 |         |
0177 |         |
0178 |         |
0179 |         |
0180
(0000) 0181 WARMSTART: EQU 0000H | Regresa al sistema operativo.
(0005) 0182 CDOS: EQU 0005H | Realiza llamada al Sistema Operativo.
(0006) 0183 BCDS: EQU 0006H | Direccion del principio del S. O.
0184
0185 |-----|
0186 |         |
0187 |         |
0188 |         |
0189 |         |
0190
0191 |-----|
0192 |         |
0193
0194 RCF: MACRO
0195 AND 7FH
0196 MEND
0197
0198 EXAF: MACRO #MENSAJE | Intercambia registros AF con AF'.
0199 EX AF,AF'
0200 MEND
0201
0202 DISPLAY: MACRO #MENSAJE
0203 LD DE,#MENSAJE
0204 CALL PUT
0205 MEND
0206
0207 DISTATUSLI: MACRO #MENSAJE
0208 LD HL,#MENSAJE
0209 LD E,00
0210 LD D,34
0211 CALL CRT
0212 MEND
0213
0214 TXMENSAJE: MACRO #MENSAJE,#LONGITUD
0215 LD IX,#MENSAJE
0216 LD B,#LONGITUD
0217 CALL CHECALIN
0218 MEND
0219
0220 |-----|
```

CROMEMCO Z80 Macro Assembler version 03.07

Sep 07, 1982 20:37:04

Pase 0006

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Origen del programa y definicion del BUFFER de E/S.

```
0100 (0100) 0222
0103 316A10 0223
0103 F3 0224 ORIGEN: ORG 10CH ; Origen del programa.
0104 114E0C 0225 ; LD SP,STACK ; Carga el Stack del programa.
0107 CD1C02 0226 ; DI ; Deshabilita interrupciones.
010A C31DOC 0227 LD DE,FLOOP
0228 CALL SETUSERC ; Cambia la direccion del salto ^C
0229 JP START ; Inicia ejecucion.
0230
0231
0232
0233
0234 IBUFFER:
0235 MAXBFSIZE: DEFB 78 ; Maxima longitud del buffer de lectura.
010D 4E 0236 BFRSIZE: DEFS 1 ; Longitud dinamica del buffer de lectura.
010E (0001) 0237 BUFFER: DEFS 78 ; Buffer de lectura.
010F (004E) 0238
```

```
0240
0241 ;*****
0242 ;*
0243 ;*   L L A M A D A S   A L   S I S T E M A   O P E R A T I V O
0244 ;*
0245 ;*****
0246
0248
0249
0250 ; Esta rutina CREA un archivo especificado en el FCB.
015D 0E16 0251 CREATE: LD C,22
015F 11D30B 0252 LD DE,FCB
0162 C30500 0253 JP CDOS
0254
0255
0256 SEARCH: LD DE,FCB
0165 11D30B 0257 LD C,17
0168 0E11 0258 JP CDOS
016A C30500 0259
0260 ; Esta rutina le cambia el nombre a un archivo del disco.
016D 11D30B 0261 RENAME: LD DE,FCB
0170 0E17 0262 LD C,23
0172 C30500 0263 JP CDOS
0264
0265 ; Esta rutina indica cual es current drive.
0175 0E19 0266 CURRENT: LD C,25
0177 C30500 0267 JP CDOS
0268
0269
0270 ; Esta rutina selecciona el drive "A" como current drive.
017A 0E0C 0271 DESELECT: LD C,12
017C C30500 0272 JP CDOS
0273
0274
0275 ; Esta rutina limpia el FCB y los apuntadores a el current drive
0276 ; ademas finaliza el programa.
017F 0E0D 0277 RESETCDOS: LD C,13
0181 C30500 0278 JP CDOS
0279
0280
0281
0282
0283 ; Esta rutina selecciona el drive especificado como current drive.
0184 0E0E 0284 SELECT: LD C,14
0186 3AD30B 0285 LD A,(FCB)
0189 FE00 0286 CP 00H
018B 2805 0287 JR Z,SELECTFIN
018D 3D 0288 DEC A
018E 5F 0289 LD E,A
018F C30500 0290 JP CDOS
0291
0192 C9 0292 SELECTFIN: RET
0293
0294
0295
0296 ; Esta rutina investiga la version de cdos.
0297 GETCDOS:
0193 0E8D 0298 LD C,141
0195 CD0500 0299 CALL CDOS
0198 79 0300 LD A,C
0199 FE52 0301 CP 52H
019B C0 0302 RET NZ
```


CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Definición de las llamadas al Sistema Operativo.

```
019C 217B01 0303 LD HL,DESELECT+1
019F 3EA2 0304 LD A,0A2H
01A1 77 0305 LD (HL),A
01A2 3A150C 0306 LD A,(FLAGS)
01A5 CBEF 0307 SET CDOS252,A
01A7 32150C 0308 LD (FLAGS),A
01AA C9 0309 RET
0310
0311 ; Esta rutina APAGA los motores de DC de los Drives.
01AB 0E76 0312 OFF: LD C,150
01AD C30500 0313 JP CDOS
0314
0315 DIRECTORIO:
01B0 0E9C 0316 LD C,156
01B2 11F40B 0317 LD DE,FCB1
01B5 C30500 0318 JP CDOS
0319
0320 ; Esta rutina ABRE un archivo especificado en el FCB.
01B8 0E0F 0321 OPEN: LD C,15
01BA 11D30B 0322 LD DE,FCB ; Direccion del File Control Block.
01BD CD0500 0323 CALL CDOS ; Llama al sistema operativo.
01C0 3C 0324 INC A ; Si el file no existe A=-1.
01C1 C0 0325 RET NZ ; Resresa si existe el archivo.
01C2 0326 DISPLAY MMSG19 ; NO SE ENCUENTRA EL ARCHIVO EN DISCO
01C8 AF 0327 XOR A ; Senala error al resreso.
01C9 C9 0330 RET ; Resresa al programa Principal.
0331
0332 ; Esta rutina cierra un archivo.
01CA 0E10 0333 CLOSE: LD C,16
01CC 11D30B 0334 LD DE,FCB ; Direccion del File Control Block.
01CF C30500 0335 JP CDOS
0336
0337 ; Esta rutina LEE el siguiente Record del archivo .
01D2 0E14 0338 READF: LD C,20
01D4 11D30B 0339 LD DE,FCB ; Direccion del File Control Block.
01D7 C30500 0340 JP CDOS
0341
0342
0343 ; Esta rutina ESCRIBE el siguiente record del archivo.
01DA 0E15 0344 WRITEF: LD C,21
01DC 11D30B 0345 LD DE,FCB ; Direccion del File Control Block.
01DF C30500 0346 JP CDOS
0347
0348
0349 ; Esta rutina BORRA el archivo del directorio.
01E2 0E13 0350 DELETE: LD C,19
01E4 11D30B 0351 LD DE,FCB ; Direccion del File Control Block.
01E7 C30500 0352 JP CDOS
0353
0354 ; Esta rutina LEE una linea de la consola del micro.
01EA 110D01 0355 GET: LD DE,IBUFFER ; Direccion del buffer.
01ED 0E0A 0356 LD C,10
01EF C30500 0357 JP CDOS
0358
0359
0360 ; Esta rutina escribe una linea de caracteres ASCII
0361 ; en la pantalla del micro.
01F2 0E09 0362 PUT: LD C,9
01F4 C30500 0363 JP CDOS
0364
0365
0366 ; Esta rutina FORMA el FCB para el archivo.
0367
```

```
0008 ;
0009 ;
0010 ; Este programa permite enlazar una macrocomputadora con
0011 ; el sistema CROMEMCO.
0012 ;
0013 ; El metodo utilizado es el siguiente:
0014 ; 1) EN MODO LOCAL.
0015 ; La microcomputadora recibe caracteres del computador
0016 ; central o del sistema microprocesador y hace "eco" en el otro
0017 ; sistema. Por ejemplo: Si la computadora central transmite,
0018 ; el microprocesador recibe el caracter y lo pone en la
0019 ; pantalla de el mismo. Si el microprocesador transmite, hace
0020 ; eco en la pantalla del caracter tecleado y lo transmite a la
0021 ; linea del computador central.
0022 ;
0023 ; 2) EN MODO DE TRANSMISION.
0024 ; El sistema microprocesador hace eco (si asi se desea) en
0025 ; la pantalla de todo aquello que se toma del disco del
0026 ; microprocesador y lo transmite al computador central.
0027 ;
0028 ; 3) EN MODO DE RECEPCION.
0029 ; Todo lo que se reciba del computador central es guardado
0030 ; en la memoria local temporalmente, para posteriormente
0031 ; guardarlo definitivamente en el disco local.
0032 ;
0033 ; Notas aclaratorias.
0034 ;
0035 ; Es importante mencionar que al TU-ART (tarjeta de
0036 ; hardware donde se conecta fisicamente el computador central)
0037 ; debe de estar conectada una linea del computador central,
0038 ; para que el sistema pueda funcionar, de lo contrario, NO
0039 ; funcionara.
0040 ;
0041 ; Asi mismo el sistema CROMEMCO debe de estar debidamente
0042 ; aterrizado, pues de otra manera se recibira basura, y el
0043 ; equipo puede danarse.
0044 ;
0045 ; Programa realizado por:
0046 ;
0047 ;
0048 ;
0049 ;
0050 ; Mario Alberto Rivera Ponce.
0051 ;
0052 ;
0053 ;
0054 ;
0055 ;
0056 ;
```

CROMEMCO Z80 Macro Assembler Version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de variables para el ensamblado.

Sep 07, 1982 20:37:04

Page 0002

```
0058 ;
0059 ;
(FFFF) 0060 TRUE EQU -1
(0000) 0061 FALSE EQU 0
0062 ;
0063 ;
0064 ;
0065 ; UNA Y SOLO UNA DE LAS SIGUIENTES VELOCIDADES DEBERAN SER VERDADERAS.
0066 ;
0067 ;
(0000) 0068 V1200: EQU FALSE ; TRUE cuando esta sea la velocidad
0069
(0000) 0070 V2400: EQU FALSE ; TRUE cuando esta sea la velocidad
0071
(FFFF) 0072 V4800: EQU TRUE ; TRUE cuando esta sea la velocidad
0073
(0000) 0074 V9600: EQU FALSE ; TRUE cuando esta sea la velocidad
0075
(FFFF) 0076 C3102: EQU TRUE ; TRUE cuando se utilice la terminal
0077 ; CROMEMCO 3102.
0078
(0000) 0079 LECTORA: EQU FALSE ; TRUE cuando solo se quiera una
0080 ; lectora de floppy.
0081
(0000) 0082 MICROS: EQU FALSE ; Cuando el enlace sea entre micros.
0083
(0000) 0084 DISKS: EQU FALSE ; Cuando se tensan discos de este tipo.
0085
(0000) 0086 MODEM: EQU FALSE ; Cuando el enlace sea a traves de modem.
0087
(FFFF) 0088 SYSZERO: EQU TRUE ; Cuando se utilice este sistema.
0089
0090
0095
0100
(00A0) 0102 VELOCIDAD: EQU 0A0H ; 4800 baud / 1 stop bit
(00FF) 0103 RETARDO: EQU 00FFH ; 0256
0105
0110
0111
0112
```

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Configuracion del TU-ART

Pase 0003

```
0114  
0115 ;  
0116 ;      *** DEFINICION DE CONTROLES DEL TU-ART ***  
0117 ;  
(0009) 0118 RESET: EQU    09H    ; Reset al Puerto de datos del Uart.  
(0002) 0119 BRK:   EQU    02H    ; Break al computador central.  
(0004) 0120 SRV:   EQU    04H    ; Break detectado desde la consola del micro.  
(0040) 0121 RDA:   EQU    40H    ; Mascara para Read data available.  
(0074) 0122 PSAR:  EQU    74H    ; Parallel Output Port.  
(0080) 0123 TBE:   EQU    80H    ; Mascara para Transmit buffer empty.  
0124  
(0000) 0125 CNTMICRO: EQU    00H    ; Puerto de control de  
0126 ;      la consola del micro.  
(0000) 0127 ABASE:  EQU    CNTMICRO ; Puerto base de la consola.  
(0001) 0128 DATMICRO: EQU    CNTMICRO+1 ; Mascara para el Puerto de  
0129 ;      datos del micro.  
0131  
(0050) 0132 CNTLINK: EQU    50H    ; Puerto de control HOST.  
0133  
0134  
0140  
(0050) 0141 PBAUD:  EQU    CNTLINK  ; Puerto de baudaje del C.C.  
(0051) 0142 DATLINK: EQU    CNTLINK+1 ; Puerto de datos del HOST.  
(0052) 0143 PCOMMAND: EQU    CNTLINK+2 ; Puerto de comandos del c.c.  
0144  
0145
```

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de caracteres ASCII.

```
0147
0148 ;*****
0149 ;*
0150 ;*
0151 ;*
0152 ;*
0153 ;*****
0154
(0000) 0155 NULL EQU 00H ; Character ASCII de NULL.
(0003) 0156 CTRLC: EQU 03H ; Character ASCII de CTRL-C.
(0005) 0157 CTRL E: EQU 05H ; Character ASCII de CTRL-E.
(0007) 0158 BEL: EQU 07H ; Character ASCII de BELL.
(0008) 0159 BS: EQU 08H ; Character ASCII de BACK SPACE.
(000D) 0160 CR: EQU 0DH ; Character ASCII de CR.
(000A) 0161 LF: EQU 0AH ; Character ASCII de LF.
(0011) 0162 DC1: EQU 11H ; Character ASCII DE X-ON.
(0012) 0163 DC2: EQU 12H ; Character ASCII de END.
(0013) 0164 CTRLS: EQU 13H ; Character ASCII de CTRL-S.
(0013) 0165 DC3: EQU 13H ; Character ASCII de x-off.
(001A) 0166 CTRLZ: EQU 1AH ; Character ASCII de CTRL-Z.
(001B) 0167 ESC: EQU 1BH ; Character ASCII de ESC.
(0020) 0168 SPACE: EQU 20H ; Character ASCII de SPACE.
(007F) 0169 DEL: EQU 7FH ; Character ASCII de DELETE.
0170
```

CROMEMCO Z80 Macro Assembler version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Definicion de localidades del sistema operativo CDOS, Y Macros.

Sep 07, 1982 20:37:04

Page 0005

```
0172
0173
0174 |-----|
0175 |          |
0176 |          |
0177 |          |
0178 |          |
0179 |-----|
0180
(0000) 0181 WARMSTART: EQU 0000H | Regresa al sistema operativo.
(0005) 0182 CDOS: EQU 0005H | Realiza llamada al Sistema Operativo.
(0006) 0183 BCDS: EQU 0006H | Direccion del principio del S. O.
0184
0185 |-----|
0186 |          |
0187 |          |
0188 |          |
0189 |-----|
0190
0191 |-----|
0192 |          |
0193 |-----|
0194 RCF: MACRO
0195 AND 7FH
0196 MEND
0197
0198 EXAF: MACRO #MENSAJE | Intercambia registros AF con AF'.
0199 EX AF,AF'
0200 MEND
0201
0202 DISPLAY: MACRO #MENSAJE
0203 LD DE,#MENSAJE
0204 CALL PUT
0205 MEND
0206
0207 DISTATUSLI: MACRO #MENSAJE
0208 LD HL,#MENSAJE
0209 LD E,00
0210 LD D,34
0211 CALL CRT
0212 MEND
0213
0214 TXMENSAJE: MACRO #MENSAJE,#LONGITUD
0215 LD IX,#MENSAJE
0216 LD B,#LONGITUD
0217 CALL CHECALIN
0218 MEND
0219
0220 |-----|
```

CROMEMCO Z80 Macro Assembler version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Origen del programa y definicion del BUFFER de E/S.

Sep 07, 1982 20:37:04

Pase 0006

```
0100 (0100) 0222
0103 316A10 0223
0103 F3 0224 ORIGEN: ORG 10CH ; Origen del programa.
0104 114E0C 0225 ; LD SP,STACK ; Carga el Stack del programa.
0107 CD1C02 0226 ; DI ; Deshabilita interrupciones.
010A C31DOC 0227 LD DE,FLOOP
0228 CALL SETUSERC ; Cambia la direccion del salto ^C
0229 JP START ; Inicia ejecucion.
0230
0231
0232
0233
010D 4E 0234 IBUFFER:
010E (0001) 0235 MAXBFSIZE: DEFB 78 ; Maxima lonsitud del buffer de lectura.
010F (004E) 0236 BFRSIZE: DEFS 1 ; Lonsitud dinamica del buffer de lectura.
0237 BUFFER: DEFS 78 ; Buffer de lectura.
0238
```

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de las llamadas al Sistema Operativo.

```
0240
0241 ;*****
0242 ;*
0243 ;*   L L A M A D A S   A L   S I S T E M A   O P E R A T I V O   *
0244 ;*
0245 ;*****
0246
0248
0249
0250 ; Esta rutina CREA un archivo especificado en el FCB.
015D 0E16 0251 CREATE: LD      C,22
015F 11D30B 0252        LD      DE,FCB
0162 C30500 0253        JP      CDOS
0254
0255
0165 11D30B 0256 SEARCH: LD      DE,FCB
0168 0E11 0257        LD      C,17
016A C30500 0258        JP      CDOS
0259
0260 ; Esta rutina le cambia el nombre a un archivo del disco.
016D 11D30B 0261 RENAME: LD      DE,FCB
0170 0E17 0262        LD      C,23
0172 C30500 0263        JP      CDOS
0264
0265 ; Esta rutina indica cual es current drive.
0175 0E19 0266 CURRENT: LD      C,25
0177 C30500 0267        JP      CDOS
0268
0269
0270 ; Esta rutina selecciona el drive "A" como current drive.
017A 0E0C 0271 DESELECT: LD      C,12
017C C30500 0272        JP      CDOS
0273
0274
0275 ; Esta rutina limpia el FCB y los apuntadores a el current drive
0276 ; ademas finaliza el programa.
017F 0E0D 0277 RESETCDOS: LD      C,13
0181 C30500 0278        JP      CDOS
0279
0280
0281
0282
0283 ; Esta rutina selecciona el drive especificado como current drive.
0184 0E0E 0284 SELECT: LD      C,14
0186 3AD30B 0285        LD      A,(FCB)
0187 FE00 0286        CP      00H
0188 2805 0287        JR      Z,SELECTFIN
018D 3D 0288        DEC      A
018E 5F 0289        LD      E,A
018F C30500 0290        JP      CDOS
0291
0192 C9 0291 SELECTFIN: RET
0292
0293
0294
0295
0296 ; Esta rutina investiga la version de cdos.
0297 GETCDOS: LD      C,141
0193 0E8D 0298        LD      C,141
0195 CD0500 0299        CALL   CDOS
0198 79 0300        LD      A,C
0199 FE52 0301        CP      52H
019B C0 0302        RET      NZ
```


CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Definición de las llamadas al Sistema Operativo.

Page 0008

```
019C 217B01      0303      LD      HL,DESELECT+1
019F 3EA2        0304      LD      A,0A2H
01A1 77          0305      LD      (HL),A
01A2 3A150C      0306      LD      A,(FLAGS)
01A5 CBEF        0307      SET    CDOS252,A
01A7 32150C      0308      LD      (FLAGS),A
01AA C9          0309      RET
0310
0311 ;          Esta rutina APAGA los motores de DC de los Drives.
01AB 0E76        0312      OFF:   LD      C,150
01AD C30500      0313      JP      CDOS
0314
0315 DIRECTORIO:
01B0 0E9C        0316      LD      C,156
01B2 11F40B     0317      LD      DE,FCB1
01B5 C30500      0318      JP      CDOS
0319
0320 ;          Esta rutina ABRE un archivo especificado en el FCB.
01B8 0E0F        0321      OPEN: LD      C,15
01BA 11D30B     0322      LD      DE,FCB ; Direccion del File Control Block.
01BD CD0500     0323      CALL   CDOS ; Llama al sistema operativo.
01C0 3C          0324      INC    A ; Si el file no existe A=-1.
01C1 C0         0325      RET    NZ ; Resresa si existe el archivo.
01C2 C0         0326      DISPLAY MMSG19 ; NO SE ENCUENTRA EL ARCHIVO EN DISCO
01C8 AF         0327      XOR    A ; Senala error al resreso.
01C9 C9         0330      RET ; Resresa al programa Principal.
0331
0332 ;          Esta rutina cierra un archivo.
01CA 0E10        0333      CLOSE: LD      C,16
01CC 11D30B     0334      LD      DE,FCB ; Direccion del File Control Block.
01CF C30500      0335      JP      CDOS
0336
0337 ;          Esta rutina LEE el siguiente Record del archivo .
01D2 0E14        0338      READF: LD      C,20
01D4 11D30B     0339      LD      DE,FCB ; Direccion del File Control Block.
01D7 C30500      0340      JP      CDOS
0341
0342
0343 ;          Esta rutina ESCRIBE el siguiente record del archivo.
01DA 0E15        0344      WRITEF: LD      C,21
01DC 11D30B     0345      LD      DE,FCB ; Direccion del File Control Block.
01DF C30500      0346      JP      CDOS
0347
0348
0349 ;          Esta rutina BORRA el archivo del directorio.
01E2 0E13        0350      DELETE: LD      C,19
01E4 11D30B     0351      LD      DE,FCB ; Direccion del File Control Block.
01E7 C30500      0352      JP      CDOS
0353
0354 ;          Esta rutina LEE una linea de la consola del micro.
01EA 110D01     0355      GET:   LD      DE,IBUFFER ; Direccion del buffer.
01ED 0E0A        0356      LD      C,10
01EF C30500      0357      JP      CDOS
0358
0359
0360 ;          Esta rutina escribe una linea de caracteres ASCII
0361 ;          en la pantalla del micro.
01F2 0E09        0362      PUT:   LD      C,9
01F4 C30500      0363      JP      CDOS
0364
0365
0366 ;          Esta rutina FORMA el FCB para el archivo.
0367
```

CROMEMCO Z80 Macro Assembler version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Definición de las llamadas al Sistema Operativo.

Sep 07, 1982 20:37:04

Page 0009

```
01F7 3A0E01    0368  FORMAT: LD      A, (BFRSIZE) ; Tamano del buffer.
01FA FE00     0369                CP      00H ; Buffer cero ????????.
01FC 2810     0370                JR      Z,FORMAT1 ; si !!, salta.
01FE 3E00     0371                LD      A,00 ; no
0200 32180C   0372                LD      (VECES),A ; restaura contador.
0203 0E86     0373  FOR:   LD      C,134
0205 11D30B   0374                LD      DE,FCB ; Direccion del File Control Block.
0208 210F01   0375                LD      HL,BUFFER ; Direccion del buffer.
020B C30500   0376                JP      CDOS
0377
020E         0378  FORMAT1: DISPLAY MMSG21 ; NO ES VALIDO NOMBRE EN BLANCO.
0214 3A180C   0381                LD      A,(VECES) ; Decrementa contador.
0217 3C       0382                INC     A ;
0218 32180C   0383                LD      (VECES),A ; /
021B C9       0384                RET ; Redresa al programa principal.
0385
021C 0E82     0386  SETUSERC:
021E C30500   0387                LD      C,130
0388                JP      CDOS
0389
0390
0391
0393
0394 ;      Esta rutina limpia la pantalla del CRT.
0395
0221 0E8E     0396  CRT:   LD      C,142
0223 C30500   0397                JP      CDOS
0398
0400
```

CROMEMCO Z80 Macro Assembler version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Definición de las llamadas al Sistema Operativo.

Sep 07, 1982 20:37:04

Page 0009

```
01F7 3A0E01    0368  FORMAT: LD      A, (BFRSIZE) ; Tamano del buffer.
01FA FE00     0369                CP      00H ; Buffer cero ????????.
01FC 2810     0370                JR      Z,FORMAT1 ; si !!, salta.
01FE 3E00     0371                LD      A,00 ; no
0200 32180C   0372                LD      (VECES),A ; restaura contador.
0203 0E86     0373  FOR:   LD      C,134
0205 11D30B   0374                LD      DE,FCB ; Direccion del File Control Block.
0208 210F01   0375                LD      HL,BUFFER ; Direccion del buffer.
020B C30500   0376                JP      CDOS
0377
020E         0378  FORMAT1: DISPLAY MMSG21 ; NO ES VALIDO NOMBRE EN BLANCO.
0214 3A180C   0381                LD      A,(VECES) ; Decrementa contador.
0217 3C       0382                INC     A ;
0218 32180C   0383                LD      (VECES),A ; /
021B C9       0384                RET ; Redresa al programa principal.
0385
021C 0E82     0386  SETUSERC:
021E C30500   0387                LD      C,130
0388                JP      CDOS
0389
0390
0391
0393
0394 ;      Esta rutina limpia la pantalla del CRT.
0395
0221 0E8E     0396  CRT:   LD      C,142
0223 C30500   0397                JP      CDOS
0398
0400
```



```

025D FE04      0466      CP      06H      ; Caracter en rango inferior
025F 380A      0467      JR      C,COMELO ; si !!, lo trasa.
0261 FE0C      0468      CP      0CH      ; ^L ?
0263 2803      0469      JR      Z,FORMFEED ; si !!, salta.
0265          0470      RCF
0267 C9         0471      RET
          0472
          0473
          0474 FORMFEED:
0268 CD6D02    0475      CALL   PAGE     ; Limpia la pantalla.
          0476
026B 37        0477      COMELO: SCF
026C C9        0478      RET           ; Regresa
          0479
026D 0E8E      0480      PAGE: LD      C,142 ; Limpia la pantalla.
026F 110000    0481      LD      DE,00   ; /
0272 CD0500    0482      CALL   CDOS     ; /
0275 C9        0483      RET
          0484
          0485
          0486 ; Esta rutina INICIA los puertos del TU-ART.
          0487 ;
0276 3E00      0488      INICIO: LD     A,ABASE ; Direccion del puerto de la consola.
0278 D374      0489      OUT    PSAR,A   ; Regresa al direccionamiento normal.
027A 3E09      0490      LD     A,RESET  ; Carga en el acumulador un RESET.
027C D352      0491      OUT    PCOMMAND,A ; Da RESET al puerto "A" del TU-ART.
027E 3EA0      0492      LD     A,VELOCIDAD ; Carga la velocidad de transmision actual.
0280 D350      0493      OUT    PBAUD,A  ; Indica velocidad de transmision al TU-ART.
0282 C9        0494      RET           ; Regresa al programa principal.
          0495
          0496
          0498
          0499
0500 ;
0501 ; Esta rutina espera un caracter de la linea del
0502 ; computador central cuando detecta un caracter del C.C. apaga
0503 ; la bandera de carry, y le quita el bit de paridad al caracter
0504 ; recibido.
0505 ; Cuando no recibe algun caracter del C.C. lee el puerto
0506 ; de datos del micro, si el caracter recibido es un <ESC>,
0507 ; significa "fin de la espera", prende la bandera de "CARRY" y
0508 ; regresa al programa Principal. Si no recibe algun caracter
0509 ; del computador central o del micro, durante aproximadamente
0510 ; 1.5 minutos, pone un aviso que dice: "TIEMPO DE RECEPCION
0511 ; EXCEDIDO". Prende la bandera de "CARRY" y regresa al
0512 ; programa Principal.
0513 ;
0283 D9        0513      LI:    EXX
0284 0632      0514      LD     B,50     ; Intercambia registros de proposito general.
0286 110100    0515      LD     DE,1     ; B' = 50
0288 210000    0516      LD     HL,0     ; DE' = 1
028C D9        0517      EXX
028E D9        0518      LI:    EXX
0290 E852      0519      SBC   HL,DE    ; Intercambia registros de proposito general.
0292 2014      0520      JR    NZ,L11   ; HL' = HL' - DE'
0294 D9        0521      EXX
0296 DB01      0522      IN    A,DATMICRO ; Si HL = 0, salta.
0298 FE1B      0523      RCF           ; Intercambia registros de proposito general.
0299 282C      0524      CP     ESC     ; Lee el puerto de datos del micro.
029B DB50      0525      JR    Z,SLI   ; Apaga el bit de paridad (bit 8).
029D E540      0526      IN    A,CNTLINK ; Compara caracter leído contra ESC
029F 282C      0527      AND   RDA     ; Lee puerto de Status del C.C.
          0528      JR    Z,L10   ; Hay dato disponible ?
          0529      ; si no lo hay, salta.

```

CROMEMCO Z80 Macro Assembler version 03.07
 C.S.C. Sistema de Transferencia de Archivos Version 04.01
 Subrutinas del Programa Principal.

Sep 07, 1982 20:37:04

Pase 0012

```

02A1 DB51      0530      IN      A,DATLINK      ;      si lo hay, lo lee
02A3          0531      RCF              ;      apaga el bit de paridad
02A5 C9        0532      RET              ;      Y regresa.
02A6 10E1     0534      LI1:  DJNZ     LIM1      ;      B = B - 1
02A8          0535      LI2:  DISPLAY MNS300   ;      TIEMPO DE RECEPCION EXCEDIDO
02AE CD2F02   0538      LI3:  CALL     CI       ;      Espera la contestacion.
02B1 20FB     0539      JR      NZ,LI3       ;      no hay nada.
02B3 FE53     0540      CP      'S'         ;      Si esperara mas
02B5 2813     0541      JR      Z,LI4       ;
02B7 FE73     0542      CP      'S'         ;
02B9 280F     0543      JR      Z,LI4       ;
02BB FE0D     0544      CP      CR          ;
02BD 280B     0545      JR      Z,LI4       ;
02BF FE4E     0546      CP      'N'         ;
02C1 2804     0547      JR      Z,SLI       ;
02C3 FE6E     0548      CP      'd'         ;
02C5 2003     0549      JR      NZ,LI4      ;
02C7 D9       0550      SLI:  EXX        ;      Intercambia registros de proposito general.
02C8 37       0551      SCF        ;      Set Carry Flag.
02C9 C9       0552      RET        ;      Regresa al programa principal.
                0553
02CA 0680     0554      LI4:  LD      B,128   ;      Prepara contador
02CC 18B5     0555      JR      LI         ;      Continua esperando.
                0556
                0558
                0559
02CE 00       0560      TESTESC:  NOP
02CF CD2F02   0561      CALL     CI
02D2 C8       0562      RET      Z
02D3 FE1B     0563      CP      ESC
02D5 C0       0564      RET      NZ
02D6 37       0565      SCF
02D7 C9       0566      RET
                0567
                0568
                0569
                0570      ;      ESTA RUTINA ESCRIBE UN TEXTO TECLADO POR EL USUARIO
0571      ;      EN LA LINEA DEL COMPUTADOR CENTRAL.
                0572
02D8 F5       0573      LOCOMP:  PUSH     AF      ;      Pone "AF" en el Stack.
02D9 3A0E01   0574      LD      A,(BFRSIZE)    ;      Carga en el acumulador el tamaño del buffer
02DC 47       0575      LD      B,A           ;      B = A
02DD F1       0576      POP     AF            ;      Rescata del stack "AF".
                0577
02DE DB50     0578      CHECALIN:  IN      A,CNTLINK ;      Lee el puerto de datos del C.C.
02E0 E680     0579      AND     TBE          ;      ¿puedo transmitir ???
02E2 28FA     0580      JR      Z,CHECALIN   ;      no, espera a que pueda.
02E4 3E00     0581      LD      A,0          ;
02E6 B8       0582      CP      B           ;      Compara B contra cero.
02E7 200A     0583      JR      NZ,CONTINUA ;      si son iguales, salta.
02E9 DD7E00   0584      LD      A,(IX+00)     ;      Carga de la memoria en el acumulador.
02EC D351     0585      OUT     DATLINK,A    ;      Saca a la línea el caracter.
02EE DD23     0586      INC     IX           ;      IX = IX + 1
02F0 05       0587      DEC     B           ;      B = B - 1
02F1 18EB     0588      JR      CHECALIN    ;      Salta a checar la línea.
                0589
02F3 3E0D     0590      CONTINUA:  LD      A,CR      ;      Carga en el acumulador un CR.
02F5 D351     0591      OUT     DATLINK,A    ;      Pone en la línea del C.C. el CR.
02F7 C9       0592      RET        ;      Regresa al programa principal.
                0593
                0594      ;
    
```

```

0595 ;          ESTA RUTINA CHECA SI TENEMOS DATOS DE LA LINEA DEL
0596 ;          COMPUTADOR CENTRAL O DEL MICRO.
0597 ;          EN CASO DE TENER DATOS DEL COMPUTADOR CENTRAL LOS ESCRIBE
0598 ;          EN EL MICRO.
0599 ;
02F8 CDE803 0600 ECO:  CALL    SCOCI          ; Lee un caracter ASCII del C.C.
02FB C8      0601      RET     Z              ; Si no hay caracter disponible, se resesa.
02FC 3A150C 0602      LD     A,(FLAGS)    ; Investida si espera un dci
02FF CB77    0603      EIT    DC?,A        ; para terminar,
0301 2007    0604      JR     NZ,ECO1    ; no, salta.
0303 3E11    0605      LD     A,DC1        ; si !! READY
0305 B9      0606      CP     C              ; son iguales ??
0306 2002    0607      JR     NZ,ECO1    ; no, salta.
0308 37      0608      SCF                    ; si !! avisa al regreso
0309 C9      0609      RET                    ; resesa.
030A CD5602 0610 ECO1:  CALL    FILTRO         ; Filtra los caracteres indeseables.
030D D42602 0611      CALL    NC,CO         ; Haz eco en el micro del caracter recibido.
0612
0613
0614
0310 C5      0615      PUSH   BC             ; Guarda el contenido de "BC" en el stack.
0311 CDE003 0616      CALL  BSRV           ; Busca un BRK desde la consola.
0314 C1      0617      POP    BC            ; Recupera el valor de "BC" del stack.
0618
0619
0620
0315 0621      RCF                    ; RCF
0317 C9      0623      RET                    ; Resesa al programa principal.
0624
0318 3E11    0625 TXDC1: LD     A,DC1        ; READY
031A CD3902 0626      CALL  LO
031D C9      0627      RET
0628
031E 3E12    0629 TXDC2: LD     A,DC2        ; END
0320 CD3902 0630      CALL  LO
0323 C9      0631      RET
0632
0324 3E13    0633 TXDC3: LD     A,DC3        ; BUSY
0326 CD3902 0634      CALL  LO
0329 C9      0635      RET
0636
032A CDCE02 0637 WDC?:  CALL    TESTESC       ;
032D 3812    0638      JR     C,WDC?1      ;
032F CDE803 0639      CALL  SCOCI         ; Espera un caracter del c.c.
0332 20F6    0640      JR     NZ,WDC?     ; no hay nada, espera.
0334 FE11    0641      CP     DC1          ; Ready ??
0336 C8      0642      RET                    ; si !!, continua.
0337 FE13    0643      CP     DC3          ; Busy ??
0339 28EF    0644      JR     Z,WDC?       ; si !!, espera.
033B FE12    0645      CP     DC2          ; End ??
033D 20EB    0646      JR     NZ,WDC?     ;
033F 37      0647      SCF                    ; si !!
0340 C9      0648      RET                    ; lo señala al regreso.
0649
0341 3A150C 0650 WDC?1: LD     A,(FLAGS)    ;
0344 CBCF    0651      SET   FIN,A         ;
0346 32150C 0652      LD     (FLAGS),A    ;
0349 C9      0653      RET                    ;
0654
0655
0656 ;          Pregunta al usuario si desea que lo recibido o lo transmitido aparezca
0657 ;          en la pantalla del micro.
0658 ;

```

```

034A 3A150C 0659 PREGUNTA:
034D CB87 0660 LD A,(FLAGS)
034F 32150C 0661 RES PANTALLA,A ; Apaga la bandera de Impresion.
0352 0662 LD (FLAGS),A
0358 CD2F02 0663 PREGO: DISPLAY MSG17 ; QUIERE TENER ECO DE LA COMUNICACION ???
035B 28FB 0664 PREG1: CALL CI ; Espera la contestacion.
035D 4F 0667 JR Z,PREG1 ; Si no se tecldea nada, trata de nuevo.
035E CD2602 0668 LD C,A ; Pone el caracter leido en el registro C.
0669 CALL CO ; Hace eco del caracter enl CRT del micro.
0670
0677
0361 FE4E 0678 CP 'N' ; Compara el acumulador contra N.
0363 2818 0679 JR Z,SPREGUNTA ; Si fueron iguales termina la rutina.
0365 FE6E 0680 CP 'n'
0367 2314 0681 JR Z,SPREGUNTA
0369 FE0D 0682 CP CR
036B 2808 0683 JR Z,PREG2
036D FE53 0684 CP 'S'
036F 2304 0685 JR Z,PREG2
0371 FE73 0686 CP 's'
0373 20DD 0687 JR NZ,PREGO
0375 3A150C 0688 PREG2: LD A,(FLAGS)
0378 CB87 0689 SET PANTALLA,A
037A 32150C 0690 LD (FLAGS),A ; Prende la bandera de Impresion.
037D C9 0691 SPREGUNTA: RET ; Resresa al programa principal.
0692
0694
0695 ; EN ESTA RUTINA SE TRANSMITE UN BREAK FISICO AL COMPUTADOR CENTRAL
0696 ; PARA ESTO SE PONE LA LINEA EN UN NIVEL ALTO POR APROXIMADAMENTE 3 MSEG
0697 ; DESPUES SE DA UN RESET AL TU-ART, Y REGRESA AL PROGRAMA PRINCIPAL.
0698 ;
037E 3E02 0699 BREAK: LD A,BRK ; Carga en el acumulador el BREAK.
0380 D352 0700 OUT PCOMMAND,A ; Pone el BREAK en el puerto de comandos.
0382 0E60 0701 LD C,60H ; Duracion del delay para los 3 Mses.
0384 0670 0702 BREAK1: LD B,70H ; Duracion del delay interior.
0386 10FE 0703 BREAK2: DJNZ BREAK2 ; B = B - 1, Si B diferente de 0 salta.
0388 0D 0704 DEC C ; C = C - 1
0389 20F9 0705 JR NZ,BREAK1 ; Si C diferente de cero, salta.
038B 3E09 0706 LD A,RESET ; Carga el el acumulador un RESET.
038D D352 0707 OUT PCOMMAND,A ; Pone el acumulador en el puerto de comandos
038F C9 0708 RET ; Resresa al programa principal.
0709
0711
0712
0390 DD215D0B 0713 INVDS: LD IX,MMSG55 ; ?DS
0394 0603 0714 LD B,03
0396 CDDE02 0715 CALL CHECALIN
0399 C9 0716 RET
0717
0718
0719
0721
0722 ; ESTA RUTINA LIMPIA LA MEMORIA DISPONIBLE PARA EL USUARIO.
0723 ;
039A CDB803 0724 LIMPIA: CALL MIDEMEM ; Mide la memoria disponible para el usuario.
039D 3E00 0725 LIMPI: LD A,NULL ; Carga en el acumulador un NULL.
039F 12 0726 LD (DE),A ; Limpia la memoria.
03A0 13 0727 DE = DE + 1
03A1 2B 0728 HL = HL - 1
03A2 7C 0729 LD A,H ; A = H
03A3 B5 0730 OR L ; HL = 00 ??
    
```


FROMMEM01 Z80 Macro Assembler version 03.07
 C.S.C. Sistema de Transferencia de Archivos Version 04.01
 Subrutinas del programa principal.

Sep 07, 1982 20:37:04

Page 0015

```

03A4 20F7      0731      JR      NZ,LIMP      ; no. salta.
03A6 C9        0732      RET              ; Regresa al programa Principal.
                0733
                0734
                0735
                0737      ;
                0738      ; de transferencia.
                0739
                0740
                0741      ; Esta rutina espera que pasen aproximadamente 3 segundos.
                0742      ;
03A7 3E12      0743      TRESEG: LD      A,18      ; A = 18
                0744      TRESEGO:
03A9 0EFF      0745      LD      C,OFFH      ; C = 256
                0746      TRESEG1:
03AB 06FF      0747      LD      B,OFFH      ; B = 256
                0748      TRESEG2:
03AD 10FE      0749      DJNZ   TRESEG2      ; B = B - 1 ; Si B = 0 hace la siguiente inst
03AF 0D        0750      DEC     C            ; C = C - 1
03B0 20F9      0751      JR      NZ,TRESEG1    ; Si C2 = 0 , salta.
03B2 3D        0752      DEC     A            ; A = A - 1
03B3 FE00      0753      CP     00           ; Compara A contra cero.
03B5 20F2      0754      JR      NZ,TRESEGO    ; Si A diferente de cero salta.
03B7 C9        0755      RET              ; Regresa al programa Principal.
                0756
                0758
                0759
0760      ; Esta rutina mide la memoria disponible para el usuario.
0761      ;
0762      MIDEMEM:
03B8 214A10    0763      LD      HL,LAST      ; HL = Direccion de la ultima Inst. del prog.
03BB E5        0764      PUSH   HL            ; Pone el stack la direccion de LAST.
03BC D1        0765      POP    DE            ; Recupera en DE la direccion de LAST.
03BD FDE5     0766      PUSH   IY            ; Pone el stack la direccion del
                0767      ; principio de CDOS.
03BF E1        0768      POP    HL            ; Recupera en HL la direccion de CDOS.
03C0 ED52     0769      SBC   HL,DE          ; HL = BCDOS - LAST
03C2 C9        0770      RET              ; Regresa al programa Principal.
                0771
                0772
0773      ; Esta rutina carga en el registro IY la direccion del principio de CDOS.
0774      ;
0775      LDBCDS:
03C3 2A0600    0776      LD      HL,(BCDOS)   ; HL = BCDOS
03C6 110001    0777      LD      DE,0100H     ; DE = PROTECCION.
03C9 ED52     0778      SBC   HL,DE          ; HL = BCDOS - 16
03CB E5        0779      PUSH   HL            ; Longitud real de memoria disponible.
03CC FDE1     0780      POP    IY            ; Longitud real de la memoria.
03CE C9        0781      RET              ; Regresa al programa Principal.
                0782
                0783
                0784      ; Esta rutina espera aproximadamente 1 Segundo.
                0785      ;
03CF 3E06      0786      UNSEG: LD      A,6      ; A = 6
03D1 0EAF     0787      UNSEG0: LD     C,0AFH   ; C = 175
03D3 06FF     0788      UNSEG1: LD     B,OFFH   ; B = 256
03D5 10FE     0789      UNSEG2: DJNZ   UNSEG2   ; B = B - 1 ; Si B = 0 continua
                0790      ; con la siguiente inst.
03D7 0D        0791      DEC     C            ; C = C - 1
03D8 20F9     0792      JR      NZ,UNSEG1     ; Si C = 0, salta.
03DA 3D        0793      DEC     A            ; A = A - 1
    
```

CROMEMCO Z80 Macro Assembler version 03.07

Sep 07, 1982 20:37:04

Pase 0016

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Subrutinas del programa principal.

```

03DB FE00 0794 CP 00 ; Compara el acumulador contra ceros.
03DD 20F2 0795 JR NZ, UNSEGO ; Si A diferente de cero, salta.
03DF C9 0796 RET ; Resresa al programa principal.
0797
0799
0800
0801 ; Esta rutina busca un BREAK desde la consola del C.C.
0802 ;
03E0 DB00 0803 BSRV: IN A,CNTMICRO ; Lee el puerto de control del micro.
03E2 E604 0804 AND SRV ; Investisa si se tecleo un BREAK.
0805
0811
03E4 CC7E03 0812 CALL Z,BREAK ; si asi fue lo transmite.
0813
0815
03E7 C9 0816 RET ; Resresa al programa principal.
0817
0818 ; Esta rutina lee un caracter del C.C.
0819 ;
03E8 DB50 0820 SCOCI: IN A,CNTLINK ; Lee el puerto de datos del comp. cent.
03EA E640 0821 AND RDA ; e investisa si hay datos disponibles
03EC C8 0822 RET Z ; si no los hay, resresa.
03ED DB51 0823 IN A,DATLINK ; Si hay un dato disponible, lo lee
03EF 0824 RCF ; y le quita el bit de paridad
03F1 4F 0826 LD C,A
03F2 C9 0827 RET ; /
0828
0829 ; Esta rutina lee un caracter del C.C. y hace eco en el micro
0830 ;
03F3 DB51 0831 LEECHAR: IN A,DATLINK ; Lee el puerto de datos de C.C.
03F5 0832 RCF ; Apasa el bit de paridad.
03F7 4F 0833 LD C,A ; Prepara el caracter para
03F8 CD2602 0836 CALL CO ; hacer eco en la pantalla.
03FB C9 0837 RET ; Resresa al programa principal.
0838
0839 ; Esta rutina sirve para funcionar como terminal del C.C.
0840 ;
03FC 00 0841 TERMINAL: NOP ; No hace nada.
03FD CDF802 0842 TER00: CALL ECO ; Hace eco de lo recibido de la linea del C.C.
0400 3817 0843 JR C,TER03
0844
0845
0852
0402 CD2F02 0853 TER01: CALL CI ; Lee un caracter de la consola del micro.
0405 29F6 0854 JR Z,TER00 ; Si no hay datos en la consola,
0855 ; los busca en el C.C.
0407 4F 0856 LD C,A ; Pone el caracter en "C".
0408 CD2602 0857 CALL CO ; Hace ECO en la pantalla del micro.
040B FE1A 0858 CP CTRLZ ; Compara el acumulador contra ^Z.
040D C8 0859 RET Z ; Si son iguales resresa.
040E FE1B 0860 CP ESC ; Compara el acumulador contra <ESC>.
0410 2002 0861 JR NZ,TER02 ; Si no fueron iguales, salta.
0412 37 0862 SCF
0413 C9 0863 RET ; aborto local, y resresa.
0414 CD3902 0864 TER02: CALL LD ; Tranite el caracter al computador central.
0417 18E9 0865 JR TER01 ; Salta para conseguir mas datos.
0866
0419 0867 TER03: RCF ; Apasa bandera de carry
041B C9 0869 RET
0870

```

```

0871
0872 ; Esta rutina lee caracteres de la consola del micro hasta un "CR"
0873 ; los guarda en el buffer, y al final graba el numero de caracteres
0874 ; que se teclearon.
0875
041C 110F01 0876 GETCON: LD DE,BUFFER ; "DE" apunta al buffer donde
0877 ; se guardan los caracteres.
041F 3E00 0878 LD A,00
0421 321001 0879 LD (BUFFER+1),A ; Limpia esta localidad de memoria.
0424 010000 0880 LD BC,0000 ; "B" es el contador de caracteres.
0881 GETCON01:
0427 CD2F02 0882 CALL CI ; Lee un caracter de la consola del micro.
042A 28FB 0883 JR Z,GETCON01 ; si no lo hay, salta hasta que lee uno.
042C FE0D 0884 CP CR ; Compara el caracter leído contra "CR"
042E 2825 0885 JR Z,GETCONSAL ; si son iguales, salta.
0430 FE08 0886 CP BS ; Compara el caracter leído contra "BS"
0432 2018 0887 JR NZ,GETCON02 ; si son diferentes, salta.
0434 3E00 0888 LD A,00 ; Carga el acumulador con ceros
0436 B8 0889 CP B ; compara el numero de
0890 ; caracteres contra ceros
0437 28EE 0891 JR Z,GETCON01 ; si el # de caracteres es cero,
0439 1B 0892 DEC DE
0893 ; si el # > 0, DE = DE - 1
043A 05 0894 DEC B ; B = B - 1
043B 0E08 0895 LD C,BS ; retrocede en la pantalla
043D CD2602 0896 CALL CO ; un caracter
0440 0E20 0897 LD C,SPACE
0442 CD2602 0898 CALL CO
0445 0E08 0899 LD C,BS
0447 CD2602 0900 CALL CO
044A 18DB 0901 JR GETCON01 ; y salta para leer mas datos
0902 GETCON02:
044C 12 0903 LD (DE),A ; El caracter leído se pone en el buffer
044D 4F 0904 LD C,A ; y hace eco en la pantalla
044E CD2602 0905 CALL CO
0451 13 0906 INC DE ; DE = DE + 1
0452 04 0907 INC B ; B = B + 1
0453 18D2 0908 JR GETCON01 ; Salta para leer otro caracter.
0909 GETCONSAL:
0455 78 0910 LD A,B ; A = B
0456 320E01 0911 LD (BFRSIZE),A ; Graba el tamaño del buffer.
0459 C9 0912 RET
0913
0914
0915
0916
0917
0918
0919
0920 ; Esta rutina transmite un ?DS al C.C.
0921 ;
0922 ;
0923 ;
0924 ;
0925 ;
0926 ;
0927 ;
0928 ;
0929 ;
0930 ;
0931 ;
0932 ;
0933 ;
0934 ;
0935 ;
0936 ;
0937 ;
0938 ;
0939 ;
0940 ;
0941 ;
0942 ;
0943 ;
0944 ;
0945 ;
0946 ;
045A CDA703 0947 INVEND: CALL TRESSEG ; Espera tres segundos para poder continuar.
045D CDA703 0948 CALL TRESSEG ; /
0460 0949 TXMENSAJE MNS632.3 , 4 ; ?END
0950
0469 C9 0951 RET ; Regresa al programa principal.
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972

```

```

0994 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0995 ;%
0996 ;%          A R E A   D E   M E N S A J E S .
0997 ;%
0998 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
0999
1000
1001
1002
046A 0D0A456C 1003 MMSG00:  DEF B CR,LF,'El tiempo de recepcion se ha excedido.'
0492 0D0A5175 1004 DEF B CR,LF,'Quieres continuar esperando, S = Si, N = No, <CS>'
04C4 24      1005 DEF B '*'
1006 ;
1007
1008
1009
04C5 0D0A4353 1017 MMSG01:  DEF B CR,LF,'CSC Sistema de transferencia de Archivos '
04F0 24      1018 DEF B '*'
1019 ;
1020
1021
04F1 0D0A4465 1022 MMSG01.1: DEF B CR,LF,'Departamento de Proyectos Especiales.'
0518 0D0A5665 1023 DEF B CR,LF,'Version 03.04 ',CR,LF
052A 24      1024 DEF B '*'
1025 ;
1026
052B 0D0A5445 1027 MMSG02:  DEF B CR,LF,'TECLEE:  CTRL-Z = Recepcion o Transmision'
0557 64652061 1028 DEF B 'de archivos.'
0563 0D0A0920 1029 DEF B CR,LF,'
058E 0D0A24    1030 DEF B CR,LF,'* CTRL-C = Regresa al Sistema Operativo.'
1031 ;
1032
0591 0D0A4375 1033 MMSG03:  DEF B CR,LF,'Cual es el nombre del archivo que quiere '
05BC 7472616E 1034 DEF B 'transmitir. ?? ',CR,LF,*
1035 ;
1036
1037
05CF 0D0A2046 1038 MMSG06:  DEF B CR,LF,' Fin de archivo.'
05E1 0D0A5175 1039 DEF B CR,LF,'Quedas en el control del programa.'
0605 0D0A0724 1040 DEF B CR,LF,BEL,*
1041 ;
1042
1043
0609 0D0A4375 1047 MMSG07:  DEF B CR,LF,'Cual es el NOMBRE del ARCHIVO en el "Disquete" ? '
063C 0D0A24    1048 DEF B CR,LF,*
1049 ;
1050
063F 0D0A456C 1050 MMSG08:  DEF B CR,LF,'El archivo ya existe.',CR,LF,*
1051 ;
0659 53652070 1052 MMSG09:  DEF B 'Se procede a su creacion.',CR,LF,*
1053 ;
1054
0675 0D0A2054 1056 MMSG12:  DEF B CR,LF,' TECLEE T = Transmision de archivos.'
069C 0D0A0920 1057 DEF B CR,LF,'
06BC 0D0A0920 1058 DEF B CR,LF,' R = Recepcion de archivos.'
06DA 0D0A24    1059 DEF B CR,LF,' D = Directorio del disco.'
1060 ;
1061
06DD 0D0A4465 1062 MMSG12.1: DEF B CR,LF,'De cual disco quieres el directorio ??'
0705 0D0A    1063 DEF B CR,LF
0707 0D0A3C45 1064 DEF B CR,LF,'<ESC> para abortar.'
071C 0D0A    1065 DEF B CR,LF
071E 0D0A2030 1066 DEF B CR,LF,' 0 = Current drive '
0732 0D0A2031 1067 DEF B CR,LF,' 1 = A '
073B 0D0A2032 1068 DEF B CR,LF,' 2 = B '
0744 0D0A203F 1069 DEF B CR,LF,' ??? <0> ',*
1070 ;
1071
0750 0D0A4E6F 1072 MMSG13:  DEF B CR,LF,'No hay espacio disponible en el directorio.,

```

```
077E 0D0A502E 1073 DEF8 CR,LF,'P. F. Usa otro disco. No puedo continuar.'
07A9 070D0A24 1074 DEF8 BEL,CR,LF,'$'
;
1075
07AD 0D0A502E 1076 MMSG14: DEF8 CR,LF,'Se excedio el tamaño máximo del archivo'
07D6 0D0A07 1077 DEF8 CR,LF,BEL
07D9 24 1078 DEF8 '$'
;
1079
07DA 0D0A4573 1080 MMSG15: DEF8 CR,LF,'Estoy grabando en el disco lo recibido ... '
0807 0D0A24 1081 DEF8 CR,LF,'$'
;
1082
080A 0D0A5265 1083 MMSG16: DEF8 CR,LF,'Recepcion cortada por el usuario.',CR,LF,'$'
;
1084
0830 0D0A5175 1086 MMSG17: DEF8 CR,LF,'Quieres tener eco de la comunicacion. ? '
085A 2C20533D 1087 DEF8 ', S= SI , N = No ! [S]? ', '$'
;
1088
0873 0D0A4E6F 1089 MMSG18: DEF8 CR,LF,'No se encuentra el archivo en el disco.'
089D 0D0A24 1090 DEF8 CR,LF,'$'
;
1091
08A0 0D0A5472 1092 MMSG19: DEF8 CR,LF,'Transmision Cortada por el Usuario.'
08C5 0D070A24 1093 DEF8 CR,BEL,LF,'$'
;
1094
08C9 0D0A4E6F 1095 MMSG21: DEF8 CR,LF,'No es valido nombre en blancos',CR,LF,BEL,'$'
;
1096
08ED 0D0A4E6F 1098 MMSG22: DEF8 CR,LF,'No hay mas espacio disponible en el disco.'
0919 0D0A 1099 DEF8 CR,LF
091B 502E462E 1100 DEF8 'P. F. Checa el tamaño de tu archivo recibido.'
0947 070D0A24 1101 DEF8 BEL,CR,LF,'$'
;
1102
094B 0D0A4375 1103 MMSG24: DEF8 CR,LF,'Cuando tengas todo listo para la transmision,'
097A 0D0A7465 1104 DEF8 CR,LF,'teclea: CTRL-Z ',CR,LF,'$'
;
1105
098F 0D0A5465 1107 MMSG26: DEF8 CR,LF,'Termina el enlace.',CR,LF
09A5 43617374 1108 DEF8 'Hasta Luego.',CR,LF,'$'
;
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119 MMSG28: DEF8 CR,LF,'Transmision abortada por el usuario. ',CR,LF,BEL,'$'
1120
1121 MMSG29: DEF8 CR,LF,'Comando no reconocido. ',CR,LF,'$'
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140 MMSG32.3: DEF8 'PEND!'
1141
1142
1143 MMSG32.4: DEF8 'RUN$(CR15)RX'
1144
1145 MMSG32.5: DEF8 'RUN$(CR15)TX'
1146
1147
1148
1149 MMSG33: DEF8 'transmision', '$'
1150
1151 MMSG34: DEF8 'recepcion', '$'
1152
1153 MMSG34.1: DEF8 'in', '$'
1154
1155 MMSG35: DEF8 CR,LF,'$'
1156
1157
1158 MMSG36: DEF8 'MOD0 TERMINAL',00H
1159
1160 MMSG36.1: DEF8 'MOD0 TERMINAL: CNTRL-Z = Menu ! CNTRL-C = '
```

CRQMCMCO Z80 Macro Assembler version 03.07

Sep 07, 1982 20:37:04

Pase 0020

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Mensajes para el usuario.

```
0A6E 46696E20 1161 DEFB 'Fin del prog.',00H
1162 ;
0A7C 44495245 1163 MMSG36.2: DEFB 'DIRECTORIO DEL DRIVE ',00H
1164
1165
0A92 4D4F444F 1166 MMSG37: DEFB 'MODO TRANSMISION ABORTAR=<ESC>',00H
1167 ;
1168 ;
0AB6 4D4F444F 1173 MMSG38: DEFB 'MODO RECEPCION',00H
1174 ;
1175 ;
0AC5 4D6F646F 1176 MMSG39.1: DEFB 'Modo recepcion, <ESC> para abortar',00H
1177 ;
0AEB 44697265 1178 MMSG38.2: DEFB 'Directorio del disco',00H
1179 ;
0AFD 4D454E55 1180 MMSG39: DEFB 'MENU DEL PROGRAMA',00H
1181 ;
0B0F 20202020 1182 MMSG40: DEFB ' ',00H
1183 ;
0B37 0D0A0D0A 1185 MMSG50: DEFB CR,LF,CR,LF,BEL,'Cualquier tecla para continuar',CR,LF,'$'
1186 ;
0B5D 3F4453 1187 MMSG55: DEFB '?DS'
1188 ;
0B60 0D0A0D0A 1189 MMSG60: DEFB CR,LF,CR,LF,BEL
0B65 496E7365 1190 DEFB 'Inserta en el drive ', '$'
1191 ;
0B7A 20756E20 1192 MMSG60.1 DEFB ' un disco inicializado con suficiente espacio disponible.'
0BB3 0D0A2070 1193 DEFB CR,LF, ' para poder seguir grabando'
0BD0 0D0A24 1194 DEFB CR,LF, '$'
1195
1196
```

```

1198 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1199 :%
1200 :%                VARIABLES Y TEMPORALES
1201 :%
1202 :%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1203
1204
OCB3  004E4E4E  1205  FCB:   DEFB 0,'NNNNNNNEEE',0,0,0,0,0 ;File control Block
1206
OCB4  0000      1207  DEFW 0,0,0,0,0,0,0,0
      0000
      0000
      0000
      0000
      0000
      0000
      0000
OCB4  003F3F3F  1208  FCB1:  DEFB 0,'?????????',0,0,0,0,0
OC05  0000      1209  DEFW 0,0,0,0,0,0,0,0
      0000
      0000
      0000
      0000
      0000
      0000
      0000
OC15  (0001)    1210
      (0000)    1211  FLAG8:  DEFS 1
      (0001)    1212  PANTALLA: EQU 0
      (0003)    1213  FIN:     EQU 1
      (0004)    1214  CAMBIO:  EQU 3
      (0005)    1215  RECEP:   EQU 4
      (0006)    1216  CDCS252: EQU 5
      (0006)    1217  DC?:     EQU 6
      1218
      1219
      1220
OC16  (0002)    1221  SAVE:    DEFS 2 ; Salva la direccion que se
      1222 ; acceso para recibir, en la memoria.
OC18  (0001)    1223  VECES:   DEFS 1 ; Contador para pedir el nombre
      1224 ; del archivo.
OC19  (0002)    1225  CONTADOR: DEFS 2 ;
OC1B  (0002)    1226  TEMPORAL: DEFS 2 ;
      1227
      1228

```

```

1230 ;*****
1231 ;*
1232 ;*          P R O G R A M A   P R I N C I P A L .
1233 ;*
1234 ;*****
1235
1236
1237
1238 START: LD      DE,00          ; DE = 00
1239        CALL    CRT           ; Limpia la pantalla del micro.
1240
1241
1242
1243
1244
1245
1246
1247        XOR     A
1248        LD      (FLAGS),A
1249        CALL    INICIO        ; Inicializa los puertos del TU-ART.
1250        CALL    GETCDOS      ; Investiga la version de cdos que se tiene.
1251
1252
1253
1254 TITULO: LD      E,0           ; E = 00
1255        LD      D,17          ; D = 17
1256        CALL    CRT           ; Comienza campo centellante.
1257        DISPLAY MMSG01       ; TITULO DE PRESENTACION DE SISTEMA
1258
1259
1260        LD      E,0           ; E = 00
1261        LD      D,18          ; D = 18
1262        CALL    CRT           ; Termina campo centellante.
1263        DISPLAY MMSG01.1     ; Continua con el titulo.
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282        LD      E,00          ; Prende la linea de "STATUS" del CRT
1283        LD      D,32          ;
1284        CALL    CRT           ; para poner en esta los comandos
1285                                ; durante el desarrollo del programa.
1286
1287
1288
1289
1290
1291
1292 FLOOP: CALL    OFF           ; Apaga los motores de DC de los DRIVES
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317 FLOOP01: LD      C,CTRLE     ; C = CTRL-E
1318        CALL    LO           ; Transmite al comp. cent. el CTRL-E.
1319
1320
1321
1322
1323
1324
1325
1326 ;
1327 ;          RUTINA PARA LA UTILIZACION DEL MICRO COMO TERMINAL DEL C.C.
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500

```


CRONEMCO Z80 Macro Assembler version 03.07
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Programa Principal.

Sep 07, 1982 20:37:04

Pase 0023

```
1335                                     ; recepción o transmisión.
0C77 260B 1336 JR Z,CON1.1 ; Si quiere salta a esa rutina.
0C78 FE03 1337 CP CTRLC ; O si quiere abortar.
0C7B CA2410 1338 JP Z,TERMINA ; Salta a esa rutina.
0C7E 4F 1339 LD C,A ; Si esta en modo de terminal,
0C7F CD3902 1340 CALL LD ; transmite el caracter al C.C.
0C82 18E9 1341 JR CONSOLE ; Busca mas datos.
1342
1343
1344 ; MENU DEL PROGRAMA.
1345
1346
0C84 00 1347 CON1.1: NOP ; No hace nada.
1348
1349
0C85 0E8E 1351 LD C,142
0C87 110000 1352 LD DE,00
0C8A CD0500 1353 CALL CD05
0C8D 1354 DISTATUSLI MSG39 ; MENU DEL PROGRAMA.
1355
1356
0C97 1362 DISPLAY MSG35 ; Pone <CR>,<LF>
0C9D 1365 DISPLAY MSG12 ; T = Tx , R = Rx
0CA3 CD2F02 1366 CON1.2: CALL CI ; Lee un caracter de la consola del micro.
0CA6 28FB 1369 JR Z,CON1.2 ; Si no se teclea nada, salta a leer de nuevo.
0CA8 4F 1370 LD C,A ; Transfiere el caracter al registro C.
0CA9 CD2602 1371 CALL CO ; Hace eco del caracter tecleado en la
1372 ; pantalla del micro.
0CAC FE52 1373 CP 'R' ; Compara el caracter con R.
0CAE CAC000 1374 JP Z,RECIBE ; Si fueron iguales, salta.
0CB1 FE72 1375 CP 'r'
0CB3 CAC000 1376 JP Z,RECIBE
0CB6 FE54 1377 CP 'T' ; Compara el caracter con T.
0CB8 2820 1378 JR Z,TRANSMITE ; Si fueron iguales, salta.
0CBA FE74 1379 CP 't'
0CBC 281C 1380 JR Z,TRANSMITE
0CBE FE44 1381 CP 'D'
0CC0 CAD00F 1382 JP Z,DIR
0CC3 FE64 1383 CP 'd'
0CC5 CAD00F 1384 JP Z,DIR
0CC8 FE1A 1385 CP CTRLZ
0CCA 28B8 1386 JR Z,CON1.1 ; Compara el caracter contra ^C
0CCC FE03 1387 CP CTRLC ; si fueron iguales termina el prog.
0CCE CA2410 1388 JP Z,TERMINA ; No se vale.
0CD1 1389 DISPLAY MSG39 ; Si no fue algo de lo comparado
0CD7 C34E0C 1392 JP FLOOP ; regresa al programa.
1393
```

```

1396
1397 ;*****$
1398 ;$
1399 ;          TRANSMISION DE ARCHIVOS.          $
1400 ;$
1401 ;*****$
1402
1403
1404 TRANSMITE:
OCDA 00      1405      NOP                ; No hace nada.
1406
1407
OCDB        1408      DISPLAY MSG33          ; Completa en la pantalla del micro:
1409
1410
1411
1412
1413
1414
1415
OCE1        1417      DISTATUSLI      MSG37 ; MODO TRANSMISION; ABORTAR = <ESC>
1418
1419
1420
1421
1422
1423
1424
OCED CD4A03  1425      CALL      PREGUNTA          ; Investida si quiere eco.
1426      TRANS00:
OCEE        1427      DISPLAY MSG03          ; Cual es el NOMBRE del ARCHIVO para Tx. ??
OCF4 CD1C04  1430      CALL      GETCON              ; Lee una linea de la consola del micro.
OCF7 CDF701  1431      CALL      FORMAT              ; Crea un "FCB" para el archivo.
OCFA 3A180C  1432      LD      A,(VECES)          ; Carga en "A" el contenido de "VECES".
OCFD FE03    1433      CP      03                  ; Investida si se teclearon tres veces.
OCFF CA4E0C  1434      JP      Z,FLOOP              ; Si fueron tres veces redresa .
OD02 FE00    1435      CP      00                  ; Investida si se teclearon blancos.
OD04 20E8    1436      JR      NZ, TRANS00         ; Si asi fue redresa a presuntar el nombre.
1437
1438
1439
OD06 CDB801  1449      CALL      OPEN                ; Abre el archivo.
OD09 2010    1450      JR      NZ,TRANS02
OD0B 3AD30B  1451      LD      A,(FCB)
OD0E 32F40B  1452      LD      (FCB1),A
OD11 CD4202  1453      CALL      CATALOG
OD14 CD4702  1454      CALL      READY
OD17 F1      1455      POP     AF
OD18 C84E0C  1456      JP      FLOOP              ; Si el archivo NO EXISTE se redresa.
OD1B 00      1457      TRANS02:
1458      NOP
1459
1460
1461
OD1C 110000  1461      LD      DE,00              ; DE = 00
OD1F CD2102  1462      CALL   CRT                ; Limpia la pantalla del CRT.
1463
1464
1465
1466
1467
1468 ;
1469 ;          RUTINA PARA PREPARAR EN EL COMPUTADOR CENTRAL LA RECEPCION.
1470
1471
1472
1473
1474
1475
OD22        1475      TXMENSAJE      MSG32.4 , 12 ; RUN RX
OD2B CDFC03  1479      CALL   TERMINAL ; Rutina para funcionar como terminal de C.C.
OD2E 2879    1480      JR      Z,CON6          ; Si la bandera de "CARRY" on aborta.
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491 ;
1492 ;          RUTINA PARA PREPARAR LA TRANSMISION EN EL MICRO.
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000

```

CROMEMCO Z80 Macro Assembler version 03.07
 C.S.C. Sistema de Transferencia de Archivos Version 04.01
 Rutina de Transmision de archivos.

Sep 07, 1982 20:37:04

Page 0025

```

0D38 CD6D02      1555          CALL    PAGE
                1556
                1557
0D38 CDD201      1558 CON2:   CALL    READF      ; Lee los primeros 128 bytes. ( 1 record).
0D3E 3D          1559          DEC     A           ; A = A - 1
0D3F 284C        1560          JR      Z,CON4     ; Si A = 0 salta.
0D41 0680        1561          LD     B,128       ; B = 80H
0D43 218000      1562          LD     HL,80H     ; HL = 80H
                1563
                1564 ;          LOOP DE TRANSMISION DEL ARCHIVO.
                1565
0D46 3E1A        1566 CON3:   LD     A,CTRLZ    ; Carga el acumulador con caracter
                1567          ; de FIN de archivo.
0D48 BE          1568          CP     (HL)       ; Compara el acumulador con CTRL-Z.
0D49 2842        1569          JR      Z,CON4     ; Si fueron iguales, salta.
                1570
                1571
0D4B 3A150C      1572 CON3.5: LD     A,(FLAGS) ; Investida si se hara eco en la pantalla.
0D4E CB47        1573          BIT   PANTALLA,A ; del caracter deseado.
0D50 C42602      1574          CALL  NZ,CO       ; Pone el caracter en la pantalla del micro.
0D53 CD3902      1575          CALL  LD          ; Transmite a la computadora central.
0D56 79          1576          LD     A,C        ; Investida si es un <LF>
0D57 F5          1577          PUSH  AF
0D58 D9          1578          EXX   ; Intercambia registros.
0D59 F1          1579          POP   AF
0D5A FE0A        1580          CP     LF         ; /
0D5C 200F        1581          JR      NZ,CON3.55 ; no, salta
0D5E CD2A03      1582          CALL  WDC?       ; si !! espera señal de listo.
0D61 3A150C      1583          LD     A,(FLAGS) ; investida si el usuario
0D64 CB4F        1584          BIT   FIN,A      ; quiere abortar
0D66 2041        1585          JR      NZ,CON6   ; si !!!
                1586
0D68 CDCF03      1587          CALL  UNSEQ      ; no espera un segundo.
0D6B 1315        1588          JR      CON3.58   ; continua.
                1589
                1590 CON3.55:
0D6D 01FF00      1591          LD     BC,RETARDO
0D70 3E00        1592          LD     A,00
0D72 B8          1593          CP     B
0D73 2803        1594          JR      Z,CON3.57
                1595
0D75 05          1596          DEC     B
0D76 20FD        1597          JR      NZ,CON3.56
                1598
0D78 3E00        1599 CON3.57: LD     A,00
0D7A E9          1600          CP     C
0D7B 2805        1601          JR      Z,CON3.58
0D7D 41          1602          LD     B,C
0D7E 0E00        1603          LD     C,00
0D80 18F3        1604          JR      CON3.56
                1605
0D82 D9          1606 CON3.58: EXX   ;
0D83 23          1607          INC   HL         ; HL = HL + 1
0D84 CDCE02      1608          CALL  TESTESC    ; Lee un caracter del puerto de datos del micro
0D87 3814        1609          JR      C,CON5   ; Si se tecleo el ESC, aborta.
0D89 10BB        1610 CON3.6: DJNZ  CON3 ; B = B - 1 ; Salta mientras es
                1611          ; diferente de cero.
0D8B 18AE        1612          JR      CON2    ; Salta para leer otro record del disco.
                1613
                1614 ;          RUTINA DE FIN DE TRANSMISION DEL ARCHIVO.
                1615
    
```

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01
Rutina de Transmision de archivos.

```
0D8D 00          1616 CON4:  NOP          ; No hace nada.
0D8E CD5A04     1617      CALL  INVEND     ; PEND
0D91 CDCA01     1618      CALL  CLOSE     ; Cierra el archivo.
                1619
                1627
0D94 11CF05     1628      LD    DE,MMSG06   ; Fin de archivo.
0D97 CDF201     1629      CALL  PUT        ; Pone el menssaje en la pantalla.
                1630
                1636
                1642
0D9A C34E0C     1643      JP    FLOOP      ; Regresa al programa principal.
                1644
                1646
                1647 ; RUTINA PARA ABORTAR LA TRANSMISION.
                1648
                1649 CONS:
                1650
                1658
0D9D          1659      DISPLAY MMSG19   ; Transmision abortada por el usuario.
0DA3 CDCA01     1662      CALL  CLOSE     ; Cierra el archivo.
                1663
                1669
0DA6 C34E0C     1670      JP    FLOOP      ; Regresa al programa principal.
                1671
                1673
                1674
                1675 ; RUTINA DE PARA ABORTAR LOCALMENTE.
                1676
0DA9 3A150C     1677      CON6:  LD    A,(FLAGS)
0DAC CB3F       1678      RES   FIN,A
0DAE 32150C     1679      LD    (FLAGS),A
0DB1          1680      DISPLAY MMSG28   ; Transmision abortada por el usuario.
                1683
                1689
0DB7 CD7E03     1690      CALL  BREAK
                1691
                1693
0DBA CDCA01     1694      CALL  CLOSE     ; Cierra el archivo.
                1695
                1701
0DBD C34E0C     1702      JP    FLOOP      ; Regresa al programa principal.
                1703
                1705
                1706
                1708
                1709
```

```

1711 :*****
1712 :$
1713 :$          RECEPCION DE ARCHIVOS.
1714 :$
1715 :*****
1716
ODC0 00      1717 RECIBE: NOP          ; No hace nada.
ODC1        1718 DISPLAY MSG034     ; Completa en la pantalla:
1719
1720
ODC7        1721
1722          1723          DISTATUSLI      MSG038 ; MODO RECEPCION
1724
1725
ODD1 CD4A03  1726          CALL      PREGUNTA     ; Investida si quiere ECO de lo recibido.
ODD4        1727 RCB01: DISPLAY MSG07     ; Cual es el nombre del archivo en el Disco.
ODDA CDEA01  1728          CALL      GET          ; Lee una linea de la consola.
ODDD CDF701  1729          CALL      FORMAT      ; Forma el File Control Block del archivo.
ODE0 3A180C  1730          LD        A,(VECES)     ; Acumulador = VECES.
ODE3 FE03    1731          CP        03          ; Compara el acumulador contra 03 .
ODE5 CA4E0C  1732          JP        Z , FLOOP     ; Si el acumulador es igual a 03,
1733          ;          redresa al programa principal.
ODE8 FE00    1734          CP        00          ; Compara el acumulador con 00 .
ODEA 20E8    1735          JR        NZ, RCB01     ; Si el acumulador diferente de 00, se redresa.
ODEC CDB801  1736          CALL      OPEN          ; Llama la rutina para abrir el archivo.
ODEF 2815    1737          JR        Z,RCB1      ; Si A = 0, salta.
ODF1        1738          DISPLAY MSG08     ; El archivo ya existe.
ODF7 3AD30B  1739          LD        A,(FCB)
ODFA 32F40B  1740          LD        (FCB1),A
ODFD CD4202  1741          CALL      CATALOG
OE00 CD4702  1742          CALL      READY
OE03 C34E0C  1743          JP        FLOOP         ; Redresa al programa principal.
OE06        1744 RCB1: DISPLAY MSG09     ; Se procede a su creacion .
OE0C CD5D01  1745          CALL      CREATE        ; Llama rutina para creacion del archivo.
OE0F 3C      1746          INC        A            ; A = A + 1
OE10 2009    1747          JR        NZ, RCB1.6    ; Si existe lugar en el directorio, salta.
OE12        1748          DISPLAY MSG13     ; Out of Entry space.
OE18 C34E0C  1749          JP        FLOOP         ; Redresa al programa principal.
1750
OE1B CD8401  1751 RCB1.6: CALL      SELECT     ; Selecciona el drive especificado
1752          ;          como el "current" drive.
OE1E CDB801  1753          CALL      OPEN          ; Llama a la rutina para abrir el archivo.
OE21 CDAB01  1754          CALL      OFF           ; Apaga los motores de DC de los drives.
OE24 CDC303  1755          CALL      LDBCDDOS      ; Llama a la rutina para cargar la
1756          ;          direccion del principio del sistema
1757          ;          operativo en la memoria.
OE27 CD9A03  1758          CALL      LIMPIA        ; Limpia toda la memoria disponible.
OE2A 3A150C  1759          LD        A,(FLAGS)
OE2D CB9F    1760          RES        CAMBIO,A
OE2F 32150C  1761          LD        (FLAGS),A
1762
OE32        1763          TXMENSAJE      MSG32.5 , 12 ; RUN TX
OE3B CDFC03  1764          CALL      TERMINAL
1765
1766
OE3E 216A10  1767          LD        HL, LAST     ; Carga la direccion de la ultima
1768          ;          instruccio. del prog.
1769
OE41        1770          DISTATUSLI      MSG38.1
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796

```

```

1797 ;      Loop de recepcion a la memoria.
1798
OE4B 0680      1799 RCB2: LD      B,128      ; Duracion del DELAY.
OE4D CD8302    1800 CALL     LI              ; Llama a la rutina para leer
                        1801                          ; un caracter del comp. cent.
OE50 2808      1802 JR       Z,RCB2.0      ; Si la bandera de CARRY "DN", salta.
OE52 4F        1803 LD      C,A           ; Pone el C el caracter leido de la linea.
OE53 FE12     1804 CP      DC2          ; FIN DEL ARCHIVO ??
OE55 200A     1805 JR      NZ,RCB2.1   ; no.
OE57 3A150C   1806 LD      A,(FLAGS)   ; si !!
OE5A CBCF     1807 RCB2.0: SET     FIN,A  ; Prende la bandera para
OE5C 3A150C   1808 LD      A,(FLAGS) ; indicar fin de
OE5F 1835     1809 JR      RCB4        ; recepcion.
                        1810
OE61 DB00     1811 RCB2.1: IN      A,CNTMICRO ; Lee el puerto de control del micro.
OE63 E604     1812 AND     SRV          ; Busca un BREAK desde la consola.
OE65 2877     1813 JR      Z,RCB6     ; Si encontro el BREAK, salta.
OE67 3A150C   1814 LD      A,(FLAGS) ;
OE6A CB47     1815 BIT     PANTALLA,A ; investisa si se hara eco de lo recibido.
OE6C C42602  1816 CALL   NZ,CO        ; Si se hara el ECO, llama a la rutina.
OE6F 79       1817 LD      A,C          ; Pone el caracter leido en el acumulador.
OE70 2A160C   1818 LD      HL,(SAVE)   ; Pone en HL
OE73 77       1819 LD      (HL),A      ; Carga el caracter leido en la memoria.
OE74 23       1820 INC     HL          ; Incrementa la direccion de la memoria.
OE75 22160C   1821 LD      (SAVE),HL ; Guarda la ultima direccion que
                        1822                          ; se puede accesar.
OE78 FE0D     1823 CP      CR          ;
OE7A 20CF     1824 JR      NZ,RCB2   ;
OE7C FE0A     1825 CP      LF          ;
OE7E 20CB     1826 JR      NZ,RCB2   ;
OE80 ED5B160C 1827 LD      DE,(SAVE)  ; DE = direccion de memoria que se puede usar.
OE84 FDE5     1828 PUSH   IX          ; Pone en el stack la direccion
                        1829                          ; del principio de CDOS.
OE86 E1       1830 POP     HL          ; Saca del stack en "HL" el principio de CDOS.
OE87 A7       1831 AND     A           ; Apaga la bandera de CARRY.
OE88 ED52     1832 SBC     HL,DE       ; HL = BCDOS - Direccion actual de mem.
OE8A 3005     1833 JR      NC,RCB3   ;
OE8C CD2403   1834 CALL   TXDC3      ; BUSY
OE8F 1805     1835 JR      RCB4        ;
OE91 CD1803   1836 RCB3: CALL  TXDC1  ; READY
OE94 18B5     1837 JR      RCB2        ; Salta para leer otro caracter
                        1838                          ; de la linea del comp. cent.
                        1839
                        1840
                        1841
1842 ;      Rutina que graba en el disco el archivo recibido.
1843
1844
OE96          1845 RCB4: DISPLAY MMSG15 ; Estoy grabando a disco todo lo recibido.
OE9C 216A10   1849 LD      HL,HLAST    ; Carga en "HL" la direccion donde esta
                        1850                          ; el principio de lo recibido.
OE9F E5       1851 PUSH   HL          ; Pone en el STACK la direccion de lo recibido.
OEAO DDE1     1852 POP    IX          ; Se recupera en IX esta direccion.
OEAA DDE5     1853 PUSH   IX          ; Pone IX en el STACK, direccion actual
                        1854                          ; para grabar.
OEAA E1       1854 POP    HL          ; HL tiene la direccion actual para grabar.
OEAS 118000   1855 LD      DE,80H     ; DE = 80H, direccion donde se deja al
                        1856                          ; sistema operativo
                        1857                          ; el record para que lo grabe.
OEAB 018000   1858 LD      BC,128    ; BC = 80H, Tamano del record.
OEAB ED80    1859 LDIR                      ; Transfiere el contenido de "HL" a "DE"

```

CRUMEMCO Z80 Macro Assembler version 03.07
 C.S.C. Sistema de Transferencia de Archivos
 Rutina de recepcion de archivos.

		1860		
OEAD	E5	1861	PUSH	HL
		1862		
OEAE	DDE1	1863	POP	IX
OEBO	CDDA01	1864	CALL	WRT
OEBS	FE02	1865	CP	02
		1866		
OEBS	2837	1867	JR	Z,R
		1868		
		1869		
		1870		
OEBS	ED5B160C	1871	RCB5.0:	LD DE,
OEBS	A7	1872	AND	A
OEBC	ED52	1873	SBC	HL,
		1874		
		1875		
OEBS	38E2	1876	JR	C,R
OECS	3A150C	1877	RCB5.1:	LD A,(
OECS	CB4F	1878	LD	FIN
OECS	280C	1879	BIT	FIN
OECS	CD0303	1880	JR	Z,R
OECA	CD9A03	1881	CALL	LDB
OECD	CD1803	1882	CALL	LIM
OECD	C34B0E	1883	CALL	TXD
		1884	JF	RCB
		1885		
		1886		
OEBS	CB6F	1887	RCB5.2:	RES FIN
OEBS	32150C	1888	LD	(FL
OEBS	CDCA01	1889	CALL	CLO
OEBS	C34E0C	1890	JP	FLO
		1891		
		1892	:	RUTINA PARA
		1893		
OEDE	00	1894	RCB6:	NOB
		1895		
		1901		
OEDE	CD7E03	1902	CALL	BRE
		1903		
		1905		
OEBS		1906	DISPLAY	MNS
OEBS	CDE201	1909	CALL	DEL
OEBS	C34E0C	1910	JP	FLO
		1911		
		1912		
		1913	:	RUTINA PARA
		1914		
OEBS	CDCA01	1915	RCB6.3:	CALL CLO
OEBS	3A150C	1916	LD	A,(
OEBS	CB5F	1917	BIT	CAM
OEBS	2036	1918	JR	NZ,I
OEBS	010C00	1919	LD	BC,I
OEBS	21D30B	1920	LD	HL,I
OEBS	11E30B	1921	LD	DE,I
OEBS	EDB0	1922	LDIR	
OEBS	010800	1923	RCB6.4:	LD BC,
OEBS	3E20	1924	LD	A,S
OEBS	21E30B	1925	LD	HL,I
OEBS	EDB1	1926	CPIR	
OEBS	2B	1927	DEC	HL
OEBS	E4C80F	1928	CALL	PD,I

CROMEMCO Z80 Macro Assembler version 03.07

Sep 07, 1982 20:37:04

Page 0030

C.S.C. Sistema de Transferencia de Archivos Version 04.01
 Rutina de recepcion de archivos.

OF11	ED5B190C	1929	LD	DE, (CONTADOR)
OF15	13	1930	INC	DE
OF16	ED53190C	1931	LD	(CONTADOR), DE
OF1A	7A	1932	RCB6.5:	LD A, D
OF1B	0630	1933	LD	B, 30H
OF1D	80	1934	ADD	A, B
OF1E	77	1935	LD	(HL), A
OF1F	23	1936	INC	HL
OF20	7B	1937	LD	A, E
OF21	80	1938	ADD	A, E
OF22	77	1939	LD	(HL), A
OF23	CD&D01	1940	CALL	RENAME
OF26	3A150C	1941	LD	A, (FLAGS)
OF29	CBDF	1942	SET	CAMBIO, A
OF2B	32150C	1943	LD	(FLAGS), A
OF2E		1944	RCB7:	DISPLAY MSG60
		1947	RCB7.0:	
OF34	3AD30B	1948	LD	A, (FCB)
OF37	FE00	1949	CP	00H
OF39	2003	1950	JR	NZ, RCB7.1
OF3B	CD7501	1951	CALL	CURRENT
OF3E	FE01	1952	RCB7.1:	CP 01H
OF40	2004	1953	JR	NZ, RCB7.2
OF42	3E41	1954	LD	A, 'A'
OF44	181A	1955	JR	RCB7.5
OF46	FE02	1956	RCB7.2:	CP 02H
OF48	2004	1957	JR	NZ, RCB7.3
OF4A	3E42	1958	LD	A, 'B'
OF4C	1812	1959	JR	RCB7.5
		1960	RCB7.3:	
OF4E	FE03	1961	CP	03H
OF50	2004	1962	JR	NZ, RCB7.4
OF52	3E43	1963	LD	A, 'C'
OF54	180A	1964	JR	RCB7.5
OF56	FE04	1965	RCB7.4:	CP 04H
OF58	2004	1966	JR	NZ, RCB7.42
OF5A	3E44	1967	LD	A, 'D'
OF5C	1802	1968	JR	RCB7.5
		1969	RCB7.42:	
OF5E	3E58	1970	LD	A, 'X'
		1971	RCB7.5:	
OF60	4F	1972	LD	C, A
OF61	CD2602	1973	CALL	C0
OF64		1974	DISPLAY	MMSG60.1
		1977	RCB7.55:	
		1978		
		1985		
OF6A	CD4702	1986	CALL	READY
OF6D	CD7A01	1987	CALL	DESELECT
OF70	CD0302	1988	CALL	FOR
OF73	010800	1989	LD	BC, 08
OF76	3E20	1990	LD	A, SPACE
OF78	21D30B	1991	LD	HL, FCB
OF7B	EDB1	1992	CP IR	
OF7D	2B	1993	DEC	HL
OF7E	E4CC0F	1994	CALL	PO, BLANO
OF81	ED5B190C	1995	LD	DE, (CONTADOR)
OF85	13	1996	INC	DE
OF86	ED53190C	1997	LD	(CONTADOR), DE
OF8A	7A	1998	LD	A, D
OF8B	0630	1999	LD	B, 30H

; Inserta en el drive

; un disco inicializado

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01 .
Rutina de recepcion de archivos.

Page 0031

OF8D	80	2000	ADD	A,B	
OF8E	77	2001	LD	(HL),A	
OF8F	23	2002	INC	HL	
OF90	7B	2003	LD	A,E	
OF91	80	2004	ADD	A,B	
OF92	77	2005	LD	(HL),A	
		2006			
		2007	RCB7.65:		
OF93	CDB801	2008	CALL	OPEN	
OF94	2808	2009	JR	Z,RCB7.7	
OF95		2010	DISPLAY	MMSG08	
OF9E	1894	2013	JR	RCB7.0	
OFA0		2014	RCB7.7: DISPLAY	MMSG09	1 SE PROCEDE A SU CREACION.
OFA6	CD5D01	2017	CALL	CREATE	
OFA9	3C	2018	INC	A	
OFAA	2008	2019	JR	NZ,RCB7.8	
OFAE		2020	DISPLAY	MMSG13	
OFB2	18B6	2023	JR	RCB7.55	
OFB4	CDB801	2024	RCB7.8: CALL	OPEN	
OFB7	CDAB01	2025	CALL	OFF	
OFBA	2A1B0C	2026	LD	HL,(TEMPORAL)	
OFBD	118000	2027	LD	DE,128	
OFD0	ED52	2028	SBC	HL,DE	
OFC2	221B0C	2029	LD	(TEMPORAL),HL	
OFC5	C3A20E	2030	JP	RCB5	
		2031			
OFC8	21EA0B	2032	NOBLAN: LD	HL,FCB+23	
OFCB	C9	2033	RET		
		2034			
OFC	21DA0B	2035	BLAN: LD	HL,FCB+7	
OFCF	C9	2036	RET		
		2037			
		2038			

```

2041 ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2042 ;!
2043 ;!          RUTINA PARA OBTENER EL DIRECTORIO DEL DISCO
2044 ;!
2045 ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2046
2047
OFD0 00          2048 DIR:   NOP
OFD1          2049 DISPLAY MMSG34.1      ; DIR
                2052
                2054
OFD7          2055 DISTATUSLI      MMSG38.2
                2060
                2062
OFE1          2063 DISPLAY MMSG12.1
OFE7 CD2F02     2066 DIR1:  CALL  CI
OFEA 28FB      2067 JR      Z,DIR1
OFE4 4F         2068 LD      C,A
OFE0 CD2602    2069 CALL   CO
OFF0 FE0B      2070 CP      CR
OFF2 281B      2071 JR      Z,DIRC
OFF4 FE1B      2072 CP      ESC
OFF6 2829      2073 JR      Z,DIR3
OFF8 FE31      2074 CP      '1'
OFFA 280A      2075 JR      Z,DIRA
OFFC FE32      2076 CP      '2'
OFFE 280B      2077 JR      Z,DIRB
1000 FE30      2078 CP      '0'
1002 280B      2079 JR      Z,DIRC
1004 18E1      2080 JR      DIR1
1006 00         2081 DIRA:  NOP
1007 3E01      2082 LD      A,1
1009 1806      2083 JR      DIR2
100B 3E02      2084 DIRB:  LD      A,2
100D 1802      2085 JR      DIR2
100F 3E00      2086 DIRC:  LD      A,0
1011 00         2087 DIR2:  NOP
1012 32F40B    2088 LD      (FCB1),A
1014          2089 DISPLAY MMSG35
1016 CD4202    2092 CALL   CATALOG
1018 CD4702    2093 CALL   READY
1021 C34E0C    2094 DIR3:  JP      FLOOP
                2095
    
```

CROMEMCO Z80 Macro Assembler version 03.07
E.S.D. Sistema de Transferencia de Archivos Version 04.01
Fin del Programa, regreso al sistema operativo.

Sep 07, 1982 20:37:04

Página 0003

```

2097 ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2098 ;
2099 ;      RUTINA  PARA  TERMINACION  DEL  PROGRAMA
2100 ;
2101 ;!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
2102
2103 TERMINA:
2104
2105
2106
1024 110000 LD      DE,0000
1027 CD2102 CALL   CRT
2108
2109
2110
2111
102A DISPLAY MMSG26 ; Fin del enlace. Hasta luego.
2112
2113
2114
2115
2116
2117
2118
2119
2120
1030 DISTATUS MMSG40 ; Limpia el status de la pantalla.
2121
2122
2123
2124
2125
2126
2127
2128
103A 1E00 LD      E,0 ; Borra la linea de status de la pantalla
103C 1621 LD      D,33 ;
103E CD2102 CALL   CRT ; /
2129
2130
2131
2132
2133
2134
2135
2136 LD      DE,0000H ; Habilita la direccion 0
1044 CD1C02 CALL   SETUSERC ; para abortar
1047 C30000 JP      WARMSTART ; Termina el programa.
2138
2139
2140
2141 ;      S T A C K   P O R   S O F T W A R E .
2142
2143 TSTACK: DEFS 32
2144 STACK:
2145 LAST: END ; Fin del programa.

Errors 0
Range Count 0

```

Symbol Table

ABASE	0000	BCDOS	0006	BEL	0007	BFRSIZE	010E	BLAND	0FCC	BREAK	037E
BREAK1	0384	BREAK2	0386	BRK	0002	BS	0008	BSRV	03E0	BUFFER	010F
C3102	FFFF	CAMBIO	0003	CATALOG	0242	CCDS	0005	CCDS252	0005	CHECALIN	02DE
CI	022F	CLOSE	01CA	CNTLINK	0050	CNTMICRO	0000	CO	0226	COMELO	0268
CON1.1	0E84	CON1.2	0CA3	CON2	0B3B	CON3	0D46	CON3.5	0D4B	CON3.55	0D4D
CON3.56	0D75	CON3.57	0D78	CON3.58	0D92	CON3.6	0B89	CON4	0D8D	CON5	0D9D
CONS	0DA9	CONSOLE	0C6D	CONTADOR	0E19	CONTINUA	02F3	CR	000D	CREATE	015D
CRT	0221	CTRLC	0003	CTRLE	0005	CTRLS	0013	CTRLZ	001A	CURRENT	0175
DATLINK	0051	DATMICRO	0001	DC1	0011	DC2	0012	DC3	001E	DC?	0006
DEL	007F	DELETE	01E2	DESELECT	017A	DIR	0FD0	DIR1	0FE7	DIR2	1011
DIRS	1021	DIRA	1006	DIRB	100B	DIRC	100F	DIRECTOR	01B0	DISK8	0000
DISPLAY	Macro	DISTATUS	Macro	ECO	02F8	EC01	030A	ESC	001B	EXAF	Macro
FALSE	0000	FCB	0BD3	FCB1	0BF4	FILTRO	0256	FIN	0001	FLAGS	0C15
FLOOP	0C4E	FLOOP01	0C65	FOR	0203	FORMAT	01F7	FORMAT1	020E	FORMFEED	0268
GET	01EA	GETCCDS	0193	GETCON	041C	GETCON01	0427	GETCON02	044C	GETCONSA	0455
IBUFFER	010D	INICIO	0276	INVS	0390	INVEND	045A	LAST	106A	LDBCDOS	03C3
LECTORA	0000	LEECHAR	03F3	LF	000A	LI	0283	LI1	02A6	LI2	02A8
LI3	02AE	LI4	02CA	LIM1	0289	LIMP	039D	LIMPIA	039A	LIO	028D
LD	0239	LOCOMP	02D8	LOOP	0C6A	MAXBFSIZ	010D	MICROS	0000	MIDMEM	0388
MNSG00	046A	MNSG01	04C5	MNSG001.1	04F1	MNSG02	052B	MNSG03	0591	MNSG06	05CF
MNSG07	0609	MNSG08	063F	MNSG09	0659	MNSG12	0675	MNSG12.1	06DD	MNSG13	0750
MNSG14	07AD	MNSG15	07DA	MNSG16	080A	MNSG17	0830	MNSG18	0873	MNSG19	08A0
MNSG21	08C9	MNSG22	08ED	MNSG24	094B	MNSG26	098F	MNSG28	09B4	MNSG29	09DF
MNSG32.3	09FB	MNSG32.4	09FF	MNSG32.5	0A0E	MNSG33	0A17	MNSG34	0A22	MNSG34.1	0A2B
MNSG35	0A2E	MNSG36	0A31	MNSG36.1	0A3F	MNSG36.2	0A7C	MNSG37	0A92	MNSG38	0AB6
MNSG38.1	0AC5	MNSG38.2	0AE8	MNSG39	0AFD	MNSG40	0B0F	MNSG50	0B37	MNSG55	0B5D
MNSG40	0B40	MNSG40.1	0E7A	MODEM	0000	NOBLAN	0FC8	NULL	0000	OFF	01AB
OPEN	01B9	ORIGEN	0100	PAGE	026D	PANTALLA	0000	PBAUD	0050	PCOMMAND	0052
PREGO	0352	PREG1	0358	PREG2	0375	PREGUNTA	034A	PSAR	0074	PUT	01F2
RCB01	0D04	RCB1	0E06	RCB1.6	0E1B	RCB2	0E4B	RCB2.0	0E5A	RCB2.1	0E61
RCB3	0E91	RCB4	0E96	RCB5	0EA2	RCB5.0	0EB7	RCB5.1	0EC0	RCB5.2	0ED3
RCB6	0EDE	RCB6.3	0EEE	RCB6.4	0F03	RCB6.5	0F1A	RCB7	0F2E	RCB7.0	0F34
RCB7.1	0F3E	RCB7.2	0F46	RCB7.3	0F4E	RCB7.4	0F56	RCB7.42	0F5E	RCB7.5	0F60
RCB7.55	0F6A	RCB7.65	0F93	RCB7.7	0FA0	RCB7.8	0FB4	RCF	Macro	RDA	0040
READF	01D2	READY	0247	READY1	024D	RECEP	0004	REC1BE	0DC0	RENAME	016D
RESET	0009	RESETCDO	017F	RETARD0	00FF	SAVE	0C16	SCODI	03E8	SEARCH	0165
SELECT	0184	SELECTFI	0192	SETUSERC	021C	SLI	02C7	SPACE	0020	SPREGUNT	037D
SRV	0004	STACK	106A	START	0C1D	SYSZERO	FFFF	TBE	0080	TEMPORAL	0C1B
TER00	03FD	TER01	0402	TER02	0414	TER03	0419	TERMINA	1024	TERMINAL	03FC
TESTESC	02CE	TITULO	0C2D	TRANS00	0CEE	TRANS02	0D1B	TRANSM(T	0CDA	TRESEG	03A7
TRESEGO	03A9	TRESEG1	03AB	TRANS03	03AD	TRUE	FFFF	TSTACK	104A	TXDC1	0318
TXDC2	031E	TXDC3	0324	TXMENSAJ	Macro	UNSEG	03CF	UNSEGO	03D1	UNSEG1	03D3
UNSEG2	03D5	V1200	0000	V2400	0000	V4800	FFFF	V9400	0000	VECES	0C18
VELOCIDA	00A0	WARMSTAR	0000	WDC?	032A	WDC?1	0341	WRITEF	01DA		

CROMEMCO Z80 Macro Assembler version 03.07		C.S.C. Sistema de Transferencia de Archivos Version 04.01		Sep 07, 1982 20:37:04	Page 0035
Symbol	Value	Defn	References		
ABASE	0000	0127	0488		
BCCDS	0006	0183	0776		
REL	0007	0158	1040 1074 1077 1093 1095 1101 1119 1185 1189		
BFRSIZE	010E	0236	0368 0574 0911		
BLAND	0FCC	2035	1294		
BREAK	037E	0699	0812 1690 1902		
BREAK1	0384	0702	0705		
BREAK2	0386	0703	0703		
BRK	0002	0119	0699		
BS	0008	0159	0826 0895 0899		
BSRV	03E0	0803	0616		
BUFFER	010F	0237	0375 0876 0879		
CS102	FFFF	0076	0392 1110 1157 1236 1252 1280 1290 1349 1415 1459 1637 1664 1696 1722 1786 2053		
CAMBIO	0003	1214	1774 1917 1942		
CATALOG	0242	0443	1453 1751 2092		
CDOS	0005	0182	0253 0258 0263 0268 0273 0279 0290 0299 0313 0318 0323 0335 0340 0347 0352 0357		
CDOS252	0005	1216	0307		
CHECALIN	02DE	0577	0580 0588 0715 0952 1478 1780		
CI	022F	0425	0451 0538 0561 0666 0853 0882 1330 1368 2066		
CLOSE	01CA	0333	1618 1662 1694 1889 1915		
CNTLINK	0050	0133	0141 0142 0143 0436 0527 0578 0820		
CNTMICRO	0000	0125	0127 0128 0415 0425 0803 1811		
CO	0226	0415	0417 0611 0669 0836 0857 0896 0898 0900 0905 1333 1371 1574 1816 1973 2069		
COMELO	026B	0477	0465 0467		
CON1.1	0C84	1347	1336 1386		
CON1.2	0CA3	1368	1369		
CON2	0D3B	1558	1612		
CON3	0D46	1566	1610		
CON3.5	0D4B	1572			
CON3.55	0D6D	1590	1581		
CON3.56	0D75	1595	1597 1604		
CON3.57	0D78	1598	1594		
CON3.58	0D82	1605	1588 1601		
CON3.6	0D89	1610			
CON4	0D8D	1616	1560 1569		
CON5	0D9D	1649	1609		
CON6	0DA9	1677	1480 1585		
CONSOLE	0C6D	1329	1341		
CONTADOR	0C19	1225	1929 1931 1995 1997		
CONTINUA	02F3	0589	0583		
CR	000D	0160	0544 0590 0682 0884 1003 1004 1017 1022 1023 1023 1027 1029 1030 1033 1034 1038		
			1039 1040 1047 1048 1050 1050 1052 1056 1057 1058 1059 1062 1063 1064 1065 1066		
			1067 1068 1069 1072 1073 1074 1076 1077 1080 1081 1083 1083 1086 1089 1090 1092		
			1093 1095 1095 1098 1099 1101 1103 1104 1104 1107 1107 1108 1119 1119 1121 1121		
			1155 1185 1185 1185 1189 1189 1193 1194 1823 2070		
CREATE	015D	0251	1757 2017		
CRT	0221	0396	1239 1256 1262 1284 1296 1302 1358 1421 1462 1728 1792 2059 2108 2128 2132		
CTRLC	0003	0156	1337 1387		
CTRLC	0005	0157	1317		
CTRLS	0013	0164			
CTRLZ	001A	0166	0858 1334 1385 1566		
CURRENT	0175	0266	1951		
DATLINK	0051	0142	0439 0530 0585 0591 0823 0832		
DATMICRO	0001	0128	0418 0428 0522		
DC1	0011	0162	0605 0625 0641		
DC2	0012	0163	0629 0645 1804		
DC3	0013	0165	0633 0643		
DC?	0006	1217	0603 1553		
DEL	007F	0169			

Symbol	Value	Defn	References
DELETE	01E2	0350	1909
DESELECT	017A	0271	0303 1987
DIR	0F50	2048	1382 1384
DIR1	0FE7	2066	2067 2080
DIR2	1011	2087	2083 2085
DIR3	1021	2024	2073
DIRA	1006	2081	2075
DIRB	100B	2084	2077
DIRC	100F	2086	2071 2079
DIRECTOR	0180	0315	0445
DISPS	0000	0084	1979
DISPLAY	Macro	0202	0326 0378 0448 0535 0663 1257 1263 1362 1365 1389 1409 1427 1659 1680 1718 1733 1746 1754 1760 1845 1906 1944 1974 2010 2014 2020 2049 2063 2089 2112
DISTATUS	Macro	0207	1292 1298 1354 1417 1724 1788 2055 2124
ECO	02F8	0690	0842 1328
ECO1	030A	0610	0604 0607
ESC	001B	0167	0525 0563 0860 2072
EYAF	Macro	0198	
FALSE	0000	0061	0068 0070 0074 0079 0082 0084 0086
FCB	0BD3	1205	0252 0256 0261 0285 0322 0334 0339 0346 0351 0374 1451 1749 1920 1921 1925 1948 1991 2032 2035
FCB1	0BF4	1208	0317 1452 1750 2088
FILTRO	0256	0462	0610
FIN	0001	1213	0651 1584 1678 1807 1878 1887
FLAG	0C15	1211	0306 0308 0602 0650 0652 0660 0662 0688 0690 1248 1552 1554 1572 1583 1677 1679 1773 1775 1806 1808 1814 1877 1888 1916 1941 1943
FLOCF	0C4E	1288	0226 1392 1434 1456 1643 1670 1702 1740 1753 1763 1890 1910 2094
FLOOP01	0C65	1316	
FOR	0203	0373	1986
FORMAT	01F7	0368	1431 1737
FORMAT1	020E	0377	0370
FORMFEED	0268	0474	0467
GET	01EA	0355	1736
GETCDOS	0193	0297	1250
GETCON	041C	0876	1430
GETCON01	0427	0881	0883 0991 0901 0908
GETCON02	044C	0902	0887
GETCONSA	0455	0909	0885
IBUFFER	0100	0234	0355
INICIO	0276	0488	1249
INVS	0390	0713	
INVEND	045A	0947	1617
LAST	106A	3145	0763 1784 1848
LDBCDOS	03C3	0775	1769 1880
LECTORA	0000	0079	0247 0342 0497 0613 0671 0720 0757 0846 0916 0954 0967 1001 1009 1026 1036 1054 1097 1124 1165 1274 1314 1324 1407 1438 1471 1620 1631 1637 1651 1664 1696 1707 2116
LEECHAR	03F3	0831	
LF	000A	0161	1003 1004 1017 1022 1023 1023 1027 1029 1030 1033 1034 1038 1039 1040 1047 1048 1050 1050 1052 1056 1057 1058 1059 1062 1063 1064 1065 1066 1067 1068 1069 1072 1073 1074 1076 1077 1080 1081 1082 1083 1086 1089 1090 1092 1093 1095 1095 1098 1099 1101 1103 1104 1104 1107 1107 1108 1119 1119 1121 1121 1155 1165 1185 1185
LI	0293	0513	0555 1800
LI1	02A6	0534	0520
LI2	02A8	0535	
LI3	02AE	0538	0539
LI4	02CA	0554	0541 0543 0545 0549
LIM1	0289	0516	0534
LIMP	039D	0725	0731
LIMPIA	039A	0724	1772 1881

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01

Page 0037

Symbol	Value	Defn	References
LIO	028D	0518	0529
LO	0239	0436	0438 0626 0630 0634 0864 1318 1340 1575
LDCOMP	02D8	0573	
LOOP	0C6A	1328	1331
MAXBFSIZ	010D	0235	
MICROS	0000	0082	1312
MIDHEM	03B9	0762	0724
MNSG00	046A	1003	0536
MNSG01	04C5	1017	1258
MNSG01.1	04F1	1022	1264
MNSG02	052B	1027	
MNSG03	0591	1033	1428
MNSG04	05CF	1038	1628
MNSG07	0609	1047	1734
MNSG08	063F	1050	1747 2011
MNSG09	0659	1052	1755 2015
MNSG12	0675	1056	1366
MNSG12.1	06DD	1062	2064
MNSG13	0750	1072	1761 2021
MNSG14	07AD	1076	
MNSG15	07DA	1080	1846
MNSG16	080A	1083	1907
MNSG17	0830	1086	0664
MNSG18	0873	1089	0327
MNSG19	08A0	1092	1660
MNSG21	08C9	1095	0379
MNSG22	08ED	1098	
MNSG24	094B	1103	
MNSG26	098F	1107	2113
MNSG28	09B4	1119	1681
MNSG29	09DF	1121	1390
MNSG32.3	09FB	1140	0950
MNSG32.4	09FF	1143	1476
MNSG32.5	0A0B	1145	1778
MNSG33	0A17	1149	1410
MNSG34	0A22	1151	1719
MNSG34.1	0A2B	1153	2050
MNSG35	0A2E	1155	1363 2090
MNSG36	0A31	1158	1293
MNSG36.1	0A3F	1160	1299
MNSG36.2	0A7C	1163	
MNSG37	0A92	1167	1418
MNSG38	0AB6	1174	1725
MNSG38.1	0AC5	1176	1789
MNSG38.2	0AE3	1178	2056
MNSG39	0AFD	1180	1355
MNSG40	0B0F	1182	2125
MNSG50	0B37	1185	0449
MNSG55	0B5D	1187	0713
MNSG60	0B60	1189	1945
MNSG60.1	0B7A	1192	1975
MODEM	0000	0086	0693 0806 1684 1896
NOBLAN	0FC8	2032	1928
NULL	0000	0155	0725
OFF	01AB	0312	1288 1768 2025
OPEN	01B8	0321	1449 1744 1767 2008 2024
ORIGEN	0100	0224	
PAGE	026D	0480	0453 0475 1555
PANTALLA	0000	1212	0561 0689 1573 1815
PBAUD	0050	0141	0493
PCOMMAND	0052	0143	0491 0700 0707

CROMEMCO Z80 Macro Assembler version 03.07

Sep 07, 1982 20:37:04

Page 0038

C.S.C. Sistema de Transferencia de Archivos Version 04.01 .

Symbol	Value	Defn	References
PREG0	0352	0663	0687
PREG1	0358	0666	0667
PREG2	0375	0688	0683 0685
PREGUNTA	034A	0659	1425 1732
PSAR	0074	0122	0489
PUT	01F2	0363	0328 0380 0450 0537 0665 1259 1265 1364 1367 1391 1411 1429 1629 1661 1682 1720
RCB01	0DD4	1733	1735 1743 1748 1756 1762 1847 1908 1946 1976 2012 2016 2022 2051 2065 2091 2114
RCB1	0E06	1754	1745
RCB1.6	0E1B	1765	1759
RCB2	0E4B	1799	1824 1826 1837 1883
RCB2.0	0E5A	1807	1802
RCB2.1	0E61	1811	1805
RCB3	0E91	1836	1833
RCB4	0E96	1845	1809 1835
RCB5	0EA2	1852	1876 2030
RCB5.0	0EB7	1871	
RCB5.1	0EC0	1877	
RCB5.2	0ED3	1887	1879
RCB6	0EDE	1894	1813
RCB6.3	0EEE	1915	1867
RCB6.4	0F03	1923	
RCB6.5	0F1A	1932	
RCB7	0F2E	1944	1918
RCB7.0	0F34	1947	2013
RCB7.1	0F3E	1952	1950
RCB7.2	0F46	1956	1953
RCB7.3	0F4E	1960	1957
RCB7.4	0F56	1965	1962
RCB7.42	0F5E	1969	1966
RCB7.5	0F60	1971	1959 1964 1968
RCB7.55	0F6A	1977	2023
RCB7.65	0F93	2007	
RCB7.7	0FA0	2014	2009
RCB7.8	0FB4	2024	2019
RCF	Macro	0194	0429 0470 0523 0531 0621 0824 0833 0867
RDA	0040	0121	0426 0528 0821
READF	01D2	0338	1558
READY	0247	0448	1454 1752 1986 2093
READY1	024D	0451	0452
RECEP	0004	1215	
RECIBE	0DC0	1717	1374 1376
RENAME	016D	0261	1940
RESET	0009	0118	0490 0706
RESETCDO	017F	0277	
RETARDO	00FF	0103	1591
SAVE	0C16	1221	1818 1821 1827 1871
SCOCI	03E8	0820	0600 0639
SEARCH	0165	0256	
SELECT	0184	0284	1765
SELECTFI	0192	0291	0287
SETUSERC	021C	0386	0227 2137
SLI	02C7	0550	0526 0547
SPACE	0020	0168	0897 1924 1990
SPREGUNT	037D	0691	0679 0681
SRV	0004	0120	0804 1812
STACK	106A	2144	0224
START	0C1D	1238	0228
SYSZERO	FFFF	0088	0130
TBE	0080	0123	0416 0437 0579
TEMPORAL	0C1B	1226	2026 2029

CROMEMCO Z80 Macro Assembler version 03.07 Sep 07, 1982 20:37:04
C.S.C. Sistema de Transferencia de Archivos Version 04.01

Symbol	Value	Defn	References
TER00	03FD	0842	0854
TER01	0402	0853	0865
TER02	0414	0864	0861
TER03	0419	0867	0843
TERMINA	1024	2103	1333 1388
TERMINAL	03FC	0841	1479 1781
TESTESC	02DE	0560	0637 1608
TITULO	0C2D	1254	
TRANS00	0CEE	1426	1436
TRANS02	0D1B	1457	1450
TRANSMIT	0CDA	1404	1378 1380
TRESEG	03A7	0743	0947 0948
TRESEGO	03A9	0744	0754
TRESEG1	03AB	0746	0751
TRESEG2	03AD	0748	0749
TRUE	FFFF	0060	0072 0076 0088
TSTACK	104A	2143	
TXDC1	0318	0625	1836 1882
TXDC2	031E	0629	
TXDC3	0324	0633	1834
TXMENSAJ	Macro	0214	0949 1475 1777
UNSEG	03CF	0786	1587
UNSEGO	03D1	0787	0795
UNSEG1	03D3	0788	0792
UNSEG2	03D5	0789	0789
V1200	0000	0068	0091
V2400	0000	0070	0096
V4800	FFFF	0072	0101
V9600	0000	0074	0106
VECES	0C18	1223	0372 0381 0383 1432 1738
VELUCIDA	00A0	0102	0492
WARMSTAR	0000	0181	2138
WDC?	032A	0637	0640 0644 0646 1582
WDC?1	0341	0650	0638
WRITER	01DA	0345	1864

3.6.- Desarrollo del programa.

El programa de comunicación está dividido en diez (10) partes; estas son:

i).- Definiciones.

a).- Para el tiempo de ensamblado.

Por medio de estas variables binarias, es posible seleccionar qué partes se incluirán en el programa que se esté diseñando en ese momento; es decir, si se desea convertir al programa en una lectora de discos para su transmisión a la computadora central, esto se especificará declarando a la variable llamada: LECTORA como cierta. De la misma manera es posible seleccionar la velocidad de comunicación con el computador central.

b).- Puerto de comunicaciones.

Aquí se define la dirección del puerto serie de la microcomputadora, por el cual se realizará el intercambio de información con el computador central. Así mismo las máscaras necesarias para el manejo de los puertos.

c).- Tablas de caracteres.

Por medio de estas tablas se definen los caracteres ASCII que se utilizan constantemente en el programa, para hacer la referencia a ellos por su nemónico y no por su valor hexadecimal.

ii).- Llamadas al sistema operativo.

Consisten en una serie de rutinas que realizan una función específica por medio de las cuales se preparan los valores necesarios para ejecutar alguna función del sistema operativo, es decir son las primitivas del sistema.

Se tienen principalmente rutinas para:

- Manejo de disco
- Manejo de pantalla.

iii).- Sub-Rutinas

Realizan funciones específicas que por utilizarse constantemente durante el programa se han incluido en esta sección. Realizan funciones muy diversas, tales como:

- Atención del puerto serie.
- Atención de la consola de operación.
- Limpieza de memoria.
- Medición de memoria.

iv).- Mensajes.

Es en este lugar donde se guardarán los mensajes con los que el sistema contestará al usuario y al computador central.

v).- Variables.

Aquí están localizadas las áreas de memoria que sirven como variables durante la ejecución del programa.

vi).- Programa principal.

a).- Inicio.

En esta parte se inicia el programa de transferencia de

archivos; así como la presentación del sistema.

b).- Terminal.

En esta sección se encuentra la parte que realiza el papel de terminal del computador central. También se detectarán los caracteres de control para entrar al menú del programa y terminar con la ejecución del mismo.

c).- Menú.

Indica las posibilidades del programa que son:

- Transmisión de archivos
- Recepción de archivos.

vii).- Transmisión.

Con este módulo se transmiten los archivos al computador central.

viii).- Recepción.

Por medio de este módulo se reciben los archivos del computador central y se graban en el disco especificado por el usuario.

ix).- Directorio del disco.

Con este módulo es posible obtener un directorio del disco que se especifique; con esto es posible identificar el archivo que se quiera transmitir; o bien verificar la no existencia de un archivo para su recepción. Todo esto sin salir del programa de comunicación.

x).- Fin del programa.

Se pone el mensaje de despedida y se devuelve el control al sistema operativo.

CAPITULO CUARTO.
CONCLUSIONES

4.- Conclusiones.

Para el desarrollo del programa de comunicación fué necesario estudiar primeramente los manuales del sistema microcomputador, así también como los del microprocesador. Esto se realizó para conocer la estructura de la máquina a utilizarse. Así también se estudiaron los manuales de las utilerías de software que permitirían el desarrollo del proyecto. Todos los manuales que no se han mencionado previamente se listan en la sección 4 de la bibliografía.

El programa de comunicación para el enlace de el microcomputador CROMEMCO con el macrocomputador BURROUGHS se realizó en forma satisfactoria, pues se lograron las metas planteadas al principio del proyecto: es decir, el programa es capaz de operar como una terminal del computador central a diferentes velocidades (estas se especifican en el programa fuente a la hora del ensamblado). Es posible también recibir archivos del computador central, grabarlos en el disco del micro bajo el nombre especificado para su consulta o acceso inmediato; transmitir un archivo que se encuentre en el disco del micro al computador central, y guardarlo en disco.

El programa de comunicación con el computador central se puede convertir en una lectora de discos. Es decir, cambiando un parámetro en el ensamblado, es posible quitar las partes que no se utilicen para la lectora de discos, pues en ésta solo se utilizan las rutinas de transmisión de archivos y la parte de la terminal. Así, también es

necesario incluir algunas otras rutinas para el funcionamiento automático de la lectora.

En las pruebas que se realizaron en el Centro de Servicios de Cómputo de la Universidad, se instaló una microcomputadora CROMEMCO conectada con el Computador Central a través de una línea directa a 9600 bauds. Se permitió a varios usuarios su utilización como terminal remota. Los resultados obtenidos fueron bastante satisfactorios, pues la velocidad de operación fue mayor que en una terminal normal.

Generalmente estas terminales están conectadas a 1200 bauds.

Se probó la recepción y transmisión de archivos. En esta parte se tuvieron algunos problemas. Los cuales se explicarán a continuación:

La parte de transmisión de archivos funcionó satisfactoriamente.

El problema mayor se presentó en el ruido que se puede introducir en la línea sobre todo a esta velocidad (9600). Al revisar el archivo transmitido en el computador central se podía encontrar que éste contenía caracteres que nunca se habían transmitido, o bien que estaban cambiados algunos caracteres del archivo. Detectar estos caracteres es una tarea muy fastidiosa o que consumía tiempo de máquina. Habría, en este último caso, que transmitir el archivo dos veces y comparar ambos archivos en el computador central para determinar cuáles son las diferencias que existen entre ambos.

En la parte de recepción de archivos se tuvieron limitantes mucho más serias, pues para recibir un archivo un poco grande es necesario partir este archivo en varios archivos de aproximadamente 80 a 150 registros, recibirlos en la microcomputadora, y posteriormente "pegarlos" todos juntos para formar el archivo original. Además nuevamente se tuvo el problema del ruido que se introduce en la transmisión.

Cuando se examina con cuidado un archivo recibido se puede detectar basura en él. Es posible la recepción doble y posteriormente la comparación de ambas versiones para determinar los posibles errores, pero nuevamente se incrementa en gran manera el tiempo de ambas máquinas y el tiempo que pierden los usuarios.

El programa de comunicación para el enlace de la BURROUGHS con una microcomputadora propuesto como tema de la presente tesis, cumple con los objetivos y metas propuestas al inicio. Al término de las pruebas se vió la necesidad de tener un control de la transferencia de información entre ambas computadoras, para esto se plantearon dos posibilidades:

i).- Desarrollar un protocolo de comunicación en ambos sistemas de cómputo, de tal manera que por medio de un programa escrito en algún lenguaje de programación de alto nivel, se tuviera en el computador Burroughs el control de la información, es decir, el transmitir caracteres de paridad, dígitos de control, caracteres de chequeos de bloque, pedir retransmisión en el caso de error, indicar recepción satisfactoria, etc.

Esto mismo se tendría que desarrollar para la microcomputadora.

ii).- Utilizar un protocolo de comunicación de datos ya implementado en Burroughs, (a nivel DCP) con el fin de desarrollar solamente este protocolo en la microcomputadora. Este protocolo de comunicación de datos deberá de indicar los errores que puedan existir en las transmisiones de información en ambos sentidos. Realizar peticiones de retransmisión si es necesario, etc.

El protocolo de comunicación de datos que realice todas las operaciones especificadas es el conocido como: "B771". Este protocolo de comunicación está implementado en el computador central, escrito en NDL, y por lo tanto corre en el DCP de Burroughs, con lo que se reduce en gran manera el tiempo de respuesta, pues de otra manera hay que esperar a que el DCP proporcione la información al programa que esté corriendo en el procesador central (caso i). Este protocolo de comunicación ya estaba probado y funcionando con los Remote Job Entry, así que sería mucho más fácil y rápido la implementación solamente para la microcomputadora.

En el apéndice 6.5 se menciona la estructura de este protocolo.

Con la implementación de el protocolo de comunicación de datos en el microcomputador, es posible tener varias líneas lógicas en una sola línea física.

Con esto, las posibilidades futuras son muy amplias. Es posible tener las líneas lógicas definidas con diferentes MCS.

Es decir, una línea o varias líneas pueden estar conectadas a CANDE, y tener la posibilidad de intercambiarse a través de ellas por software.

Así, es factible poder manejar procesos en paralelo. Esta opción será recomendable, solo que se proporcione a determinados usuarios en los casos que el trabajo que realizan así lo requiera.

Otra de las líneas lógicas puede estar definido al MCS SYSTEM/IMPRESORA por medio del cual es posible recibir la impresión de algún programa ó compilación de la Burroughs y esta impresión realizarla en la impresora del microcomputador.

Por último, es recomendable la utilización del programa de comunicación con la Burroughs para aplicaciones en donde el trabajo mayor se efectúe como una terminal remota, pues

existe la misma probabilidad de que se introduzca ruido en un mensaje si se está trabajando con una terminal, que si se está trabajando con una microcomputadora. En caso de que el usuario observe que en cierto momento se introduce basura en sus mensajes, él pedirá al computador central la retransmisión de la o las líneas del programa que observe con error. De esta manera el computador central no se saturará en el DCP, pues el tener una línea declarada sin ningún tipo de protocolo o chequeo consume menos recursos, y además tampoco consume muchos recursos del computador central.

Para aplicaciones en donde se quiera transmitir una gran cantidad de información al computador central. Es decir, la aplicación principal es una lectora de discos. Se recomienda la implantación del programa de comunicación, con un protocolo de comunicación para evitar que la información que se está transmitiendo tenga error, pues si se trata de un programa fuente, al momento de la compilación se detectarán los errores. Sin embargo, si se trata de un archivo de datos, por ejemplo, una nómina, los errores serían demasiado caros, y esto ocasionaría muchos problemas. Pues un dígito cambiado en el RFC o bien en el saldo deudor o acreedor de cada cheque sería un gran error.

Pretendemos continuar esta investigación, pues sería muy interesante y además útil realizar una versión con un protocolo que fuera capaz de atender a varios usuarios simultáneamente (solo funcionando como terminales del computador central). Posteriormente tener un Procesador de Entrada/Salida en donde se esté corriendo continuamente el programa de protocolo. Cuando se tenga algo para el usuario del microcomputador, este deberá interrumpir al usuario para indicarle que existe un mensaje del computador central.

En el aspecto económico, de un enlace de este tipo, puede verse como algo demasiado caro y difícil de afrontar:

es necesario adquirir el equipo del microcomputador, con su unidad de discos. Si se desea utilizar el microprocesador a toda su capacidad, es conveniente adquirir la terminal que soporte el fabricante, así mismo como una impresora para tenerla en forma local.

Todo este equipo tiene un costo aproximado de \$ 10,000.00 dólares, que al cambio de 100 pesos por 1 dólar hace un total de \$ 1,000,000.00 de pesos. Además el mantenimiento tiene un costo de 1.2 % mensual sobre el precio del equipo.

Por otra parte veamos los costos de una terminal "tonta" conectada al equipo BURROUGHS B-6700:

1 hora de CPU \$ 36,000.00	1 minuto de CPU \$ 600.00
1 hora de E/S \$ 27,000.00	1 minuto de E/S \$ 450.00

Una sesión promedio de teleproceso, tiene una duración de 75 minutos, lo que da un total de 1500 minutos mensuales, con un costo de \$ 675,000.00.

Ahora supongamos que este tipo de enlace disminuya el tiempo de teleproceso en un 25 % (que es una cifra muy conservadora); esto significa que se ahorra \$168,750 pesos cada mes, con lo que el microcomputador quedara pagado en 6 ó 7 meses.

Cabe señalar que con este equipo, además de comunicarse con el computador central, será posible desarrollar programas de aplicación en lenguajes de alto nivel (ver capítulo 3). así mismo podría manejarse el procesamiento de palabra, pues se está considerando la adquisición de las impresoras adecuadas.

Con esto queda demostrado que es más barato tener una

micromcomputadora conectada a un gran sistema central, que además puede operar en forma local, que tener solamente terminales "tontas" conectadas al computador central, y realizar en éste todo el trabajo.

Como consideración final, habría que señalar, que esfuerzos como el descrito en el desarrollo de esta tesis serían mucho más fructíferos si pudieran establecerse bajo normas de observancia general en el ámbito universitario. Sin embargo, recomendamos la implantación de normas en el área de cómputación dentro de la Universidad, con el fin de poder aprovechar al máximo trabajos como el aquí descritos.

Consideramos que la ampliación y uso de lo ya logrado, son las consecuencias naturales que deberán redundar en el beneficio de la comunidad Universitaria. Si esto se lograra, el trabajo aquí descrito quedaría ampliamente justificado.

CAPITULO QUINTO
BIBLIOGRAFIA

- 2.1.- BURROUGHS B6700
Information Processing Systems
Reference Manual
- 2.2.- DOBIN ROSENTHAL, BORIS
Interconexión entre dos computadoras B-6700.
Tesis Profesional.
Facultad de Ciencias
México D.F. 1979
- 2.3.- N.C.R. Data Communications Concepts.
Blacksburg
- 2.4.- Salgueiro Ensueta, Emilio
Tesis Profesional.
Facultad de Ingeniería.
México D.F. 1981
- 2.5.- DATA COMMUNICATIONS HANDBOOK
Signetics
San Francisco 1980.
- 2.6.- Network Data Language
Burroughs 1981
- 2.7.- Data Communicator

BURROUGHS
October 26, 1970

2.8.- Cande
BURROUGHS
August 1977

3.1.- INTERFACE AGE
January 1981
Vol. 6, Issue 1

3.2.- CROMEMCO ZPU
Z-80 CENTRAL PROCESSING UNIT
INSTRUCTION MANUAL
January 1980.

3.3.- CROMEMCO 16FDC
FLOPPY DISK CONTROLLER
INSTRUCTION MANUAL
August 1980

3.4.- CROMEMCO 64KZ
RANDOM ACCESS MEMORY
INSTRUCTION MANUAL
January 1980

3.5.- CROMEMCO TU-ART
DIGITAL INTERFACE
INSTRUCTION MANUAL
January 1980

3.6.- CROMEMCO Z80A CENTRAL PROCESSING UNIT
TECNICAL MANUAL

July 1980

3.7.- ZAKS, RODNAY

HOW TO PROGRAM THE Z80

Third Edition

Radio Shack

3.8.- CROMEMCO CDOS

INSTRUCTION MANUAL

June 1981

3.9.- CROMEMCO MACRO ASSEMBLER

INSTRUCTION MANUAL

October 1980

3.10.- CANNON, DON L. PH. D.

LUCKE, GERALD. M SEE

UNDERSTANDING MICROPROCESSORS

TEXAS INSTRUMENT LEARNING CENTER

1979

3.11.- CAMPOS ESTRADA, HECTOR

Diseño y desarrollo de un sistema de
Bases de datos distribuidas vía un
microcomputador.

Tesis Profesional

Facultad de Ingeniería.

(En proceso).

3.12.- COLMENARES BARON, RAUL

Estudio de viabilidad para la implantación de
Microprocesadores de apoyo en la Universidad.

Abril de 1981.

- 4.1.- SCREEN EDITOR
CROMEMCO
Instruction Manual
January 1981

- 4.2.- EDIT
CROMEMCO
July 1979

- 4.3.- Texas Instrument OMNI 800
Serial printer
1980

- 4.4.- CROMEMCO SYSTEM THREE
Instruction manual
June 1979

- 4.5.- CROMEMCO SYSTEM ZERO
Instruction manual
February 1981

- 4.6.- CROMEMCO 3102

- 4.7.- CROMEMCO DM10
Technical user Manual
Remote video display terminal
Beehive International
January 1980

- 4.8.- CROMEMCO SINGLE CARD COMPUTER
Instruction manual
January 1980

4.9.- CROMEMCO TM10

Technical manual

April 1981

6.1.- HOGAN, THOM

OSBORNE CP/M USER GUIDE

OSBORNE/McGraw-Hill

Berkley, California.

6.2.- GARETS, MARK

INTERFACING S-100/IEEE 696 MICROCOMPUTERS

OSBORNE/McGraw-Hill

Berkley, California.

6.5.- BURROUGHS

REMOTE JOB ENTRY (RJE) SYSTEM

Information Manual

C A P I T U L O S E X T O
A P E N D I C E S

- Apendice I

CP/M

INTRODUCCION.

CP/M es un sistema operativo para microcomputadoras, producido por una compañía llamada Digital Research. Este sistema operativo fué diseñado para utilizarse en sistemas de microcomputadoras basadas en un microprocesador 8080. CP/M son las siglas de "Control Program/Monitor".

Estructura de CP/M

Cuando este sistema operativo es cargado en la memoria de la computadora, CP/M por sí mismo se carga en la parte más alta del bloque de memoria libre, el cual generalmente es de 7,168 bytes (7 Kbytes) de longitud. También, dos instrucciones de salto son puestas en la parte baja de la memoria. La primera es puesta en la localidad 0000H y es llamada "Warm Start Vector". La segunda está localizada en 0005H y se llama: FDIS Entry Vector. El usuario puede utilizar toda la memoria entre 0100H y la parte baja de CP/M, dirección que está especificada en FDIS. Después de cargar el sistema operativo, la memoria contiene lo siguiente:

Localidad de Memoria (Hexadecimal)	Localidad de Memoria (HEX)	Función
0000	C3	Warm Start Vector
0001	03	
0002	xx	
0003	00	IOBYTE (usualmente)
0004	00	Drive Seleccionado
0005	C3	FDDS Vector de Ent.
0006	06	
0007	yy	
0008-005B		Areas reservadas Interrupciones y scratch
005C-007C		File Control Block por default
007D-007F		Posición del Registro Aleatorio por default
0080-00FF		(CP/M 2.0) Buffer del Disco por default
0100-(yy00-1)		(TPA) Transient Program Area. Lugar donde los programas son ejecutados
yy00-zzFF		Sistema Operativo CP/M (contiene varias secciones: CCP, BCLOS Y BIOS)

Tabla que muestra la distribución de la memoria en el sistema operativo CP/M.

Nota: xx,yy,zz son utilizados aquí como substitutos para

el valor de la dirección de memoria en hexadecimal, varía dependiendo del tamaño de la memoria disponible y de la versión de CP/M que se esté utilizando.

Otra manera de describir la estructura de CP/M es la siguiente:

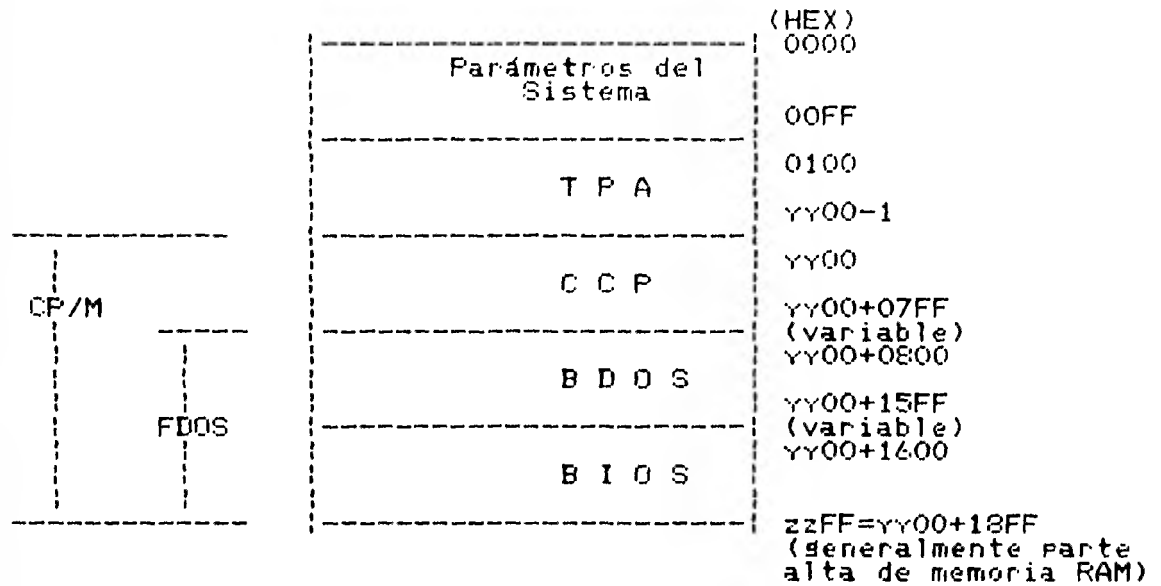


Diagrama de la estructura del sistema operativo CP/M.

Esta representación simplificada de la estructura de CP/M no está dibujada a escala; la TPA es la sección más grande de la memoria aproximadamente 58 Kbytes de memoria en los sistemas de 64 K.

CCP CONSOLE COMMAND PROCESSOR (Procesador de Comandos de la Consola)

El módulo CCP interpreta los comandos de CP/M que el usuario teclea. Esta etapa del CP/M generalmente solo es importante cuando se vé en la pantalla el prompt (carácter utilizado por una computadora o programa para indicar que está lista y esperando algún evento por parte del usuario). El CCP reconoce seis o siete comandos de CP/M; así como

también es capaz de reconocer algunos caracteres especiales, los llamados subcomandos.

Cuando el CCP no reconoce el comando del usuario, este verifica el directorio de archivos en el disco por un archivo ejecutable (*.COM). Si encuentra este programa, lo carga en la memoria y transfiere el control a este programa.

BDOS BASIC DISK OPERATING SYSTEM (Sistema Operativo Básico)

Toda la actividad del disco pasa a través de esta sección del CP/M. BDOS no se puede acceder a través de comandos directos de la consola. Para esto el CCP o el programa que se está ejecutando pone el número de la llamada deseada en el registro "C" y después se ejecuta una llamada a la localidad 0005 de la memoria.

Entre otras funciones BDOS contiene las siguientes:

- Sistema de Reset al Disco
- Selección del Drive
- Creación de Archivos
- Apertura de Archivos
- Cerrado de Archivos
- Búsqueda en el Directorio
- Borrado de Archivos
- Cambio de nombre a los Archivos

- Lectura de Archivos Aleatorios o Secuenciales

- Escritura de Archivos Aleatorios o Secuenciales

- Investigar el espacio disponible en el disco

- Investigar el Drive seleccionado

- Indicadores de Archivos Set/Reset

Además cuenta con algunas rutinas lógicas de la consola, lectora, perforadora, e impresora.

Si una función utiliza varios parámetros estos son proporcionados a través de los registros "DE". Algunas funciones regresan información al programa que lo llamó a través de los registros de CPU, esto depende de la rutina llamada.

BIOS THE BASIC INPUT/OUTPUT SYSTEM (El Sistema Basico de Entrada/Salida)

Esta es la parte que el usuario puede confeccionar a su gusto, dependiendo de la configuración de la microcomputadora que tenga. Esta parte es proporcionada por el vendedor, no por Digital Reseach. Las funciones de BIOS incluyen:

- Lectura de un carácter de la consola

- Escritura de un carácter a la consola

- Lectura de un carácter del dispositivo de lectura

- Perforado de un carácter al dispositivo de perforación
- Escritura de un carácter al dispositivo de impresión
- Control del estado de I/O
- Escritura de una cadena de caracteres a la consola
- Presuntar a la consola acerca de su estado.

El procedimiento de Cold Start.

Cuando el usuario carga el sistema (boot), el controlador del disco mueve el cargador de cold start a la memoria. El cargador es ejecutado. Este carga a CP/M a la memoria y lo coloca en la localidad adecuada en la parte alta de la memoria RAM disponible.

Una vez que CP/M ha sido cargado a la memoria, el control se pasa a ejecutar la primera rutina en la sección de BIOS de CP/M, la cual es: la rutina de cold start. Esta rutina inicia todas las localidades de memoria que utilizan los dispositivos periféricos que trabajarán bajo CP/M, después de esto despliega un mensaje en la pantalla para informar al operador que el sistema operativo ha sido cargado satisfactoriamente.

Por último, el control es pasado al CCP, ésta inicia algunas pocas localidades de memoria, y finalmente responde con el prompt, para esperar los comandos del usuario. El CCP recolecta los caracteres tecleados por el usuario y los

procesa hasta recibir un carriage return.

Podemos ver que la estructura básica de CDOS es el CP/M por medio de esta breve explicación podemos comprender el funcionamiento de CDOS el cual es uno de los pilares fundamentales del sistema de transferencia de archivos descrito en esta tesis.

* referencia 6.1

- Apendice II

Bus S-100.

La palabra BUS proviene del Latin "Omnibus" que significa: para todos.

Todas la tarjetas de CROMEMCO que se encuentran en el mercado, están diseñadas para poder efectuar fácilmente el acoplamiento entre la tarjeta deseada y el bus estandar industrial S-100. A este bus originalmente se le denominaba como el bus "Altair", el cual aparece en la línea MITS Altair de computadoras en 1975. El bus pronto fué adoptado por una gran parte de los fabricantes de computadoras en todo el mundo y fué denominado Bus S-100 por el Dr. Roger Melen de la compañía CROMEMCO en agosto de 1976, pues este bus utiliza 100 líneas para la transmisión de señales.

Fisicamente el bus S-100 consiste en una serie de 100 contactos tipo peine, montados en una tarjeta llamada "mother board" y alambrados en paralelo. Los módulos o tarjetas que se insertan en el bus S-100 tienen las siguientes dimensiones: 5" por 10" pulgadas.

El bus S-100 originalmente fué diseñado para utilizarse con un módulo de CPU utilizando el microprocesador 8080, y por consecuencia la definición de las señales siguen muy de cerca a las del sistema 8080.

Las señales del microprocesador Z-80A difieren de las del 8080 (por ejemplo: el Z-80A es manejado por un reloj de una sola fase, mientras el 8080 es manejado por un reloj de dos fases).

Las señales del bus están agrupadas en ocho diferentes

categorías, y estas son:

1.- Direcciones	16 o 24
2.- Datos	16
3.- Estados	9
4.- Control de salida	5
5.- Control de entrada	6
6.- Interrupciones	12
7.- Control TNA	8
8.- Utilería	22

El sistema S-100 es muy popular debido a las ventajas que ofrece a sus usuarios. Aquí presentamos algunas de ellas.

1.- Independencia del Procesador.

Actualmente existen fabricantes de ocho diferentes tarjetas de CPU's de 8 bits, estos son: 8080A, 8085, Z80, 2650, 6502, 6800, 6802 y 6809 y sistemas diferentes de tarjetas de CPU's de 16 bits como son: 8086, 8088, Z8000, 68000 y la micromáquina de Pascal; todas diseñadas para sistema S-100. Ningún otro sistema de cómputo ofrece tal variedad de arquitecturas de CPU.

2.- Software Disponible.

Existen varios lenguajes disponibles, sistemas operativos; así como paquetes de aplicación para sistemas basados en S-100.

3.- Soporte de Hardware.

Existen cerca de 100 diferentes fabricantes de cerca de 400 tarjetas diferentes que se pueden utilizar con este sistema.

4.- Gran capacidad de cómputo.

Los sistemas S-100 ofrecen la capacidad más grande de cómputo, que cualquier otro sistema de microcomputadora. Ningún otro sistema de microcomputadoras ofrece acceso directo a más de 16 Mbytes de memoria (con 24 líneas de direcciones); hasta 10 líneas para vectores de interrupción.

5.- Estandarización.

El bus S-100 está completamente estandarizado; esto provee un grado muy alto de compatibilidad entre las diferentes tarjetas listas para conectarse de los diferentes fabricantes.

Las señales del bus S-100 pueden ser agrupadas en cuatro categorías funcionales: 1) fuentes de poder, 2) direcciones, 3) datos, 4) reloj y señales de control. Una lista completa de las señales del bus S-100 se muestran a continuación:

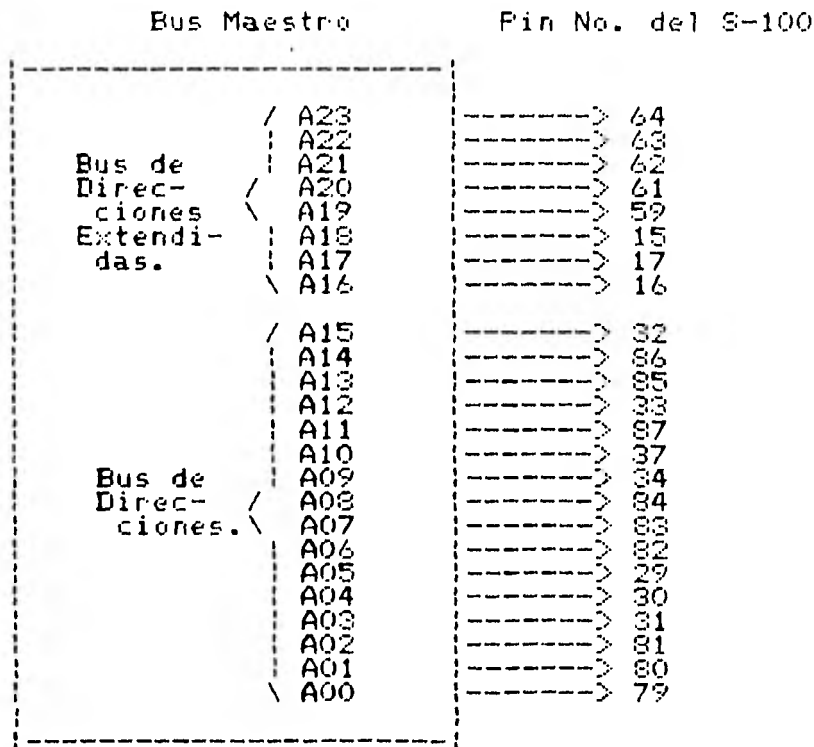
1.- +8 VOLTS	26.- PHLDA	51.- +8 VOLTS	76.- PSYNC
2.- +18 VOLTS	27.- PWAIT	52.- -18 VOLTS	77.- PWR
3.- EXT. READY	28.- PINTE	53.- SSW DISABLE	78.- PDBIN
4.- NO DEFINIDA	29.- A5	54.- CLEAR EXT.	79.- A0
5.- NO DEFINIDA	30.- A4	55.- NO DEFINIDA	80.- A1
6.- NO DEFINIDA	31.- A3	56.- NO DEFINIDA	81.- A2
7.- NO DEFINIDA	32.- A15	57.- NO DEFINIDA	82.- A7
8.- NO DEFINIDA	33.- A12	58.- NO DEFINIDA	83.- A8
9.- NO DEFINIDA	34.- A9	59.- NO DEFINIDA	84.- A8
10.- NO DEFINIDA	35.- D01	60.- NO DEFINIDA	85.- A13
11.- NO DEFINIDA	36.- D00	61.- NO DEFINIDA	86.- A14
12.- NMI	37.- A10	62.- NO DEFINIDA	87.- A11
13.- NO DEFINIDA	38.- D04	63.- NO DEFINIDA	88.- D02
14.- NO DEFINIDA	39.- D05	64.- NO DEFINIDA	89.- D03
15.- NO DEFINIDA	40.- D06	65.- MEM. REQUEST	90.- D07
16.- NO DEFINIDA	41.- D12	66.- REFRESH	91.- D14
17.- NO DEFINIDA	42.- D13	67.- NO DEFINIDA	92.- D15
18.- STAT. DISAB.	43.- D17	68.- MEM. WRITE	93.- D16
19.- CONT. DISAB.	44.- sM1	69.- PROTECT STAT.	94.- D11
20.- UNPROT. MEM	45.- sQUT	70.- PROTECT MEM.	95.- D10
21.- SINGLE STEP	46.- sINP	71.- RUN	96.- sINTA
22.- ADDR. DISAB.	47.- sMEMR	72.- PREADY	97.- sWD
23.- DO DISAB.	48.- sHLTA	73.- PINT	98.- sSTACK/4MH
24.- o2 CLOCK	49.- 2MHZCLK	74.- PHOLD	99.- PWR-ON CLR
25.- o1 CLOCK	50.- GROUND	75.- PRESET.	100.- GROUND

Tabla que muestra las conexiones físicas en una tarjeta del bus S-100.

A continuación se describen los diferentes "buses" que componen el bus S-100.

El bus de Datos.

El bus de datos, que se muestra en la siguiente figura, es utilizado por el sistema para determinar en donde están localizados la memoria y los dispositivos de E/S en el "espacio de direcciones".



El bus de direcciones y el bus de direcciones extendidas.

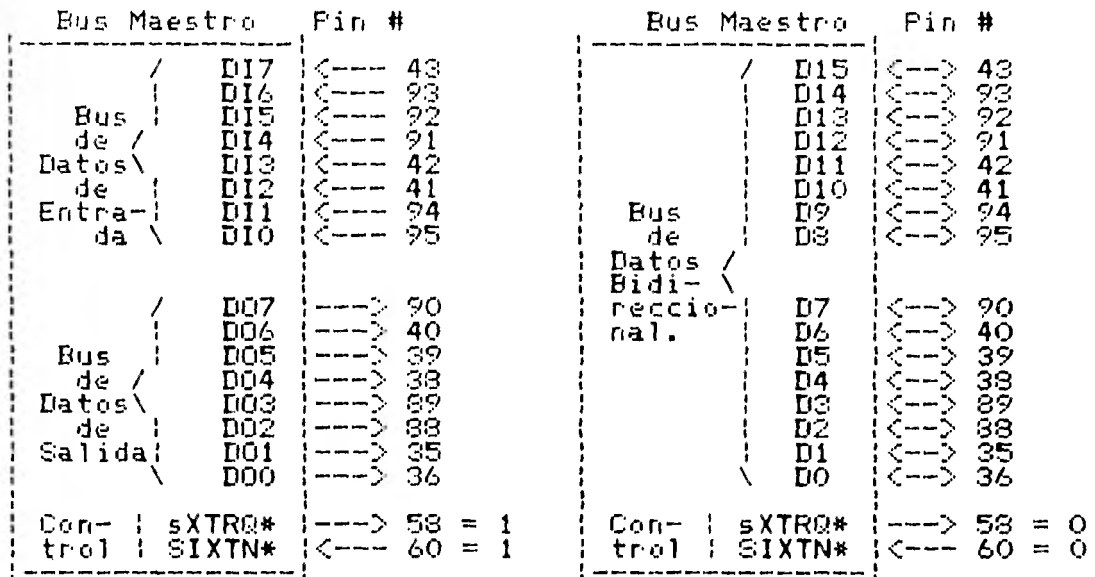
El antiguo diseño del bus S-100 tiene 16 líneas de direcciones para memoria (A0-A15). El nuevo estándar especifica ocho líneas de memoria adicionales, haciendo con esto un total de 24 (A0-A23). Con 16 líneas de direcciones, el total de memoria direccionada era de 65,536 bytes. Ahora el total es de 16,777,216 bytes, o aproximadamente 16 Megabytes.

El antiguo bus S-100 también utilizaba ocho líneas para las direcciones de E/S (A0-A7), y por lo tanto podía direccionar solamente 256 puertos de E/S. La versión actualizada establece que el sistema debe de utilizar 16 líneas de E/S (A0-A15), direccionando con esto un número de

puertos de E/S de 65,536.

El bus de datos.

El sistema S-100 posee dos buses de ocho bits cada uno, como se muestra en la siguiente figura. El primero es llamado "bus de datos de salida", y el segundo "bus de datos de entrada". La dirección del flujo es siempre relativa a la corriente maestra. Por lo tanto, el bus de datos de salida podrá transportar datos del dispositivo principal hacia el dispositivo secundario, (tal como cuando se escribe en la memoria), y el bus de entrada de datos podrá tener datos que van del dispositivo secundario al dispositivo principal (como cuando se lee de memoria).



Buses unidireccionales de 8 bit cuando:
 sXTRQ* = 1 y
 SIXTN* = 1

Bus bidireccional de 16 bits cuando:
 sXTRQ* = 0 y
 SIXTN* = 0

Figura que muestra el bus de datos del S-100.

Quando se diseñó el bus S-100, solo se tomaron en cuenta los CPU's de 8 bits. Pero el estándar acepta los CPU's de 16 bits. En este caso el bus de entrada de datos, y el bus de salida de datos son combinados para formar un bus bidireccional de 16 bits de ancho. Esto significa que los datos pueden circular en ambas direcciones del bus (la dirección depende del tipo de ciclo que se esté efectuando).

Bus de estados.

Existen ocho líneas de estados en el bus S-100. De la información contenida en estas líneas, cualquier dispositivo en el sistema, puede determinar que tipo de ciclo de bus el dispositivo principal se encuentra efectuando. Por ejemplo, la línea sMEMR es una señal de estado que indica que el ciclo actual es una lectura a la memoria. Un dispositivo secundario puede ver a las líneas de estado y direcciones para determinar si debe de responder en ese momento.

Los nemónicos para las líneas de estados empiezan con una letra minúscula 's'. Debido a que el bus S-100 fue originalmente utilizado con el microprocesador 8080 de Intel, por lo que muchas de los nemónicos de las líneas de estado y control provienen de la hoja de datos de Intel. Las líneas de datos son:

MEMR	Memory read
MI	Op-code fetch
OUT	Output
INP	Input
WD*	Memory or output write
INTA	Interrupt acknowledge
HLTA	Halt acknowledge
XTRQ*	16-bit data transfer request

Bus de Control de Salida.

El bus de control de salida está constituido por un grupo de señales que le indican al sistema cuando hacer las cosas. Estas señales son conocidas como "strobes". Estas difieren de las señales anteriores (direcciones, datos, estados) pues estas indican qué información es la que se está utilizando. Por ejemplo, durante un ciclo de lectura de memoria este bus contiene una dirección que especifica qué tarjeta de memoria es la que se está accedendo, el bus de estados contiene información para indicar que este no es un ciclo de lectura de memoria (es decir, que tipo de ciclo), pero la línea de control del bus de salida PDBIN le dice a la tarjeta de memoria cuando poner los datos en el bus.

Existen cinco señales de control de salida, en todos estos su nemónico comienza con la letra minúscula 'p', estos son:

PSYNC	Indica principio de un nuevo ciclo de bus
PDBIN	Lee strobe (pulso)
PWR*	Escribe strobe (pulso)
PHLDA	Hold acknowledge
PSTVAL*	Dirección y estado estable en el bus.

Todas las señales son generadas por el bus maestro actual, con la excepción de PWR* y SOUT, las cuales son usualmente generadas por un circuito en la tarjeta permanente principal.

Bus de Control de Entrada.

Las seis señales en el bus de control de entrada son generadas por dispositivos secundarios para indicar al dispositivo principal actual que debe de hacer algo, (esperar por ejemplo). Las dos señales, RDY y XRDY, indican precisamente esto. Ellas le indican al bus maestro que debe de extender el ciclo de bus actual.

La señal HOLD* le indica al bus permanente que pare lo que está haciendo y ceda el control a otro bus principal. La señal INT* es una petición de interrupción de un dispositivo secundario al dispositivo principal, mientras NMI* es similar, pero es una petición de interrupción no enmascarable.

La última señal de control de entrada es SIXTN*. Esta señal le indica al bus principal actual que el dispositivo secundario actual puede recibir 16 bits de datos. Esta señal se origina localmente en cada uno de los diferentes

dispositivos secundarios.

Bus de Interrupciones.

Existen 10 líneas en el S-100 utilizadas para interrumpir al dispositivo principal su tarea actual y ponerlo a realizar otra tarea. Esto se utiliza principalmente en sistemas de multiusuarios, o en aplicaciones de tiempo real. Las interrupciones pueden ser generadas por cualquier dispositivo secundario.

El dispositivo que quiera "mirar" a las interrupciones, usualmente puede ignorarlas por medio de una técnica conocida como "enmascaramiento", pero existe una interrupción que no puede ser enmascarada y es llamada NMI* (siglas de Non-Maskable Interrupt). NMI* usualmente se utiliza para dar servicio en casos prioritarios tales como una posible falla del suministro eléctrico.

Bus de Control de TMA.

Las señales de control TMA controlan la transferencia del bus permanente principal a uno principal temporal y viceversa. TMA significa Temporary Master Access (Aceso Temporal Principal).

Bus Auxiliar.

Las líneas de las fuentes de tensión y todas las otras señales del bus que realmente no las podemos catalogar dentro de las categorías anteriores, están agrupadas juntas al

final. Sin embargo estas señales son muy importantes para la operación del sistema. RESET* es una de estas señales, junto con SLAVE CLR* y POC* (Power-On-Clear).

También está incluida en esta categoría la señal del reloj del sistema. Todos los tiempos del sistema S-100 son relativos a este reloj maestro del sistema. Otras señales son: CLOCK, cuando se utiliza un reloj de 2MHZ; MWRT, un pulso para la escritura a memorias; y PHANTOM* que es una señal que habilita o deshabilita una memoria ROM, esto se utiliza en el inicio del sistema.

FUENTES DE TENSION DEL BUS S-100

+8 Volts	Conectores 1 y 51
+18 Volts	Conector 2
-18 volts	Conector 52
Tierra	Conectores 50 y 100

Tres fuentes de tensión aparecen en el bus S-100: +8 volts, +18 volts y -18 volts. Las fuentes de tensión principales no están reguladas, así que la regulación de la tensión se deberá de efectuar en cada una de las tarjetas que estén conectadas al bus S-100, usualmente esto se realiza por reguladores de tres terminales, del tipo 78xx. La regulación distribuida de la tensión, tiene varias ventajas sobre la regulación centralizada:

-Cada tarjeta está protegida individualmente de una sobre carga de tensión. Una falla en la fuente central de tensión no puede destruir todo el sistema de computación.

-El calor producido por los reguladores de voltaje, se distribuye más uniformemente en un volumen mayor.

-Las caídas de tensión a lo largo del bus, no influyen en el voltaje que recibe cada una de las tarjetas del bus.

-El costo inicial de la infraestructura es menor.

* referencia 3.2 y 6.2

- Apéndice III

Z80.

El CPU del Z-80 contiene 208 bits de memoria de Lectura/Escritura) que son accesibles al programador. Esta memoria se encuentra distribuida de la siguiente manera: 16 registros de 8 bits cada uno y cuatro registros de 16 bits cada uno; esto lo podemos representar de la siguiente manera:

Registros Principales		Registros Alternos	
ACUMULADOR A	BANDERA F	ACUMULADOR A'	BANDERA F'
B	C	B'	C'
D	E	D'	E'
H	L	H'	L'
REGISTROS DE PROPOSITO GENERAL			

Vector de Interrupciones	I	REGISTROS DE PROPOSITO ESPECIAL
Refrescamiento de Memoria	R	
Registro de Indice	IX	
Registro de Indice	IY	
Apuntador a la pila	SP	
Contador del programa	PC	

Todos los registros del Z-80 se encuentran implementados por RAM estático. Los registros incluyen dos conjuntos de 6 registros de propósito general que pueden ser utilizados individualmente como registros de 8 bits, o bien en pares como registros de 16 bits.

Los registros de propósito especial son los siguientes:

i.- Contador de Programa (Program Counter PC)

El PC contiene la dirección de 16 bits de la instrucción que se encuentra ejecutando, (es la dirección de la instrucción que carga la memoria). El PC se incrementa automáticamente después de que su contenido ha sido transferido a las líneas de dirección. Cuando ocurre una transferencia de control incondicional (JUMP), el nuevo valor es colocado automáticamente en el PC.

ii.- Apuntador a la pila (Stack Pointer SP)

El SP contiene la dirección (en 16 bits) del

actual tope de la pila. Esta pila o stack se maneja por software en cualquier tipo de memoria RAM externa. Esta pila está organizada como una pila LIFO (Last-In First-Out); es decir, lo último en entrar a la pila es lo primero en salir de ella. Las direcciones o datos pueden ser introducidos o retirados de la pila por medio de las instrucciones PUSH Y POP respectivamente.

La utilización permite la implementación de interrupciones de niveles múltiples, anidamiento ilimitado de subrutinas y la simplificación de muchos tipos de manipulación de datos.

iii.- Registros Indexados (IX, IY).

Estos registros indexados contienen una dirección (en 16 bits) que es utilizada en los modos de direccionamiento indexado. En este modo, un índice de registro es utilizado como base para apuntar a una región de memoria de la cual el dato debe ser almacenado o recuperado (por ejemplo un buffer). Un byte adicional es incluido en las instrucciones indexadas para especificar un desplazamiento a partir de esta base. Este desplazamiento es especificado como un entero con signo en complemento a dos. Este modo de direccionamiento simplifica muchos tipos de programas, especialmente aquellos donde son usadas tablas de datos.

iv.- Registro de la dirección de Interrupción de Página (Interrupt Page Address Register: I)

El CPU del Z-80 puede ser operado en un modo en que una llamada indirecta a cualquier localidad de memoria puede ser obtenida en respuesta a una interrupción. El registro I es utilizado con este propósito para almacenar los 8 bits de orden mayor de la dirección indirecta, mientras el dispositivo interrumpido provee los 8 bits de menor orden de la dirección. Esta forma permite interrumpir rutinas para ser localizadas dinámicamente en cualquier lugar en memoria, con un tiempo de acceso mínimo para la rutina.

v.- Registro de Refrescamiento de Memoria
(Memory Refresh Register: R).

El CPU del Z-80 contiene un contador de refrescamiento de memoria para permitir que las memorias dinámicas sean utilizadas con la misma facilidad que las memorias estáticas. Este registro consta de 8 bits, de los cuales 7 son incrementados automáticamente después de que cada instrucción es cargada. El último bit permanece programado como el resultado de una instrucción del tipo:

LD R,A ; Carga R con el acumulador.

El dato en el contador de refrescamiento es sacado en la porción menor del bus de direccionamiento junto con una señal de control de refrescamiento, mientras el CPU decodifica y ejecuta la instrucción cargada. Este modo de refrescamiento es totalmente transparente al

programador y no disminuye la velocidad de operación del CPU. El programador puede cargar el registro R para propósitos de pruebas, pero este registro no es muy usado. Durante el refrescamiento el contenido del registro I es colocado en los 8 bits superiores del bus de dirección.

- Acumulador y registros de banderas.

El CPU incluye 2 acumuladores independientes de 8 bits cada uno y registros asociados de banderas de 8 bits también. El acumulador contiene los resultados de operaciones lógicas o aritméticas de 8 bits, indicando si el resultado de una operación es, ó no igual a cero. De la misma manera se puede preguntar por el resultado de las siguientes operaciones: Paridad Par, Paridad Impar, Acarreo, Medio-Acarreo y signo. El programador selecciona a voluntad el acumulador y el par de banderas con que desea trabajar con una simple instrucción de intercambio, de tal forma que puede trabajar fácilmente con cada par.

- Unidad aritmética y lógica.

Las instrucciones aritméticas y lógicas del CPU son ejecutadas en la ALU (Arithmetic Logical Unit Unidad Aritmética-y-Lógica). La ALU se comunica internamente con los registros y el bus de datos interno. Las funciones que provee la ALU son las siguientes:

- Suma
- Resta
- And lógico
- Or lógico
- Xor lógico
- Comparaciones
- Corrimientos y rotaciones a la derecha e izquierda
- Incrementos
- Decrementos
- Enciende un bit
- Reset bit
- Prueba un bit

También existen dos (2) conjuntos semejantes de registros de propósito general. Cada conjunto contiene 6 registros de 8 bits, cada uno, que pueden ser utilizados individualmente como registros de 8 bits, o como pares de registros de 16 bits. Un conjunto de los anteriores es llamado BC, DE y HL, mientras que el conjunto complementario es llamado: BC', DE' y HL'. El programador puede seleccionar cualquier conjunto de registros para trabajar a través de un solo comando para el conjunto entero. Estos registros tienen muchas aplicaciones especialmente en los sistemas que están basados en memorias ROM, pues es a través de estos donde se

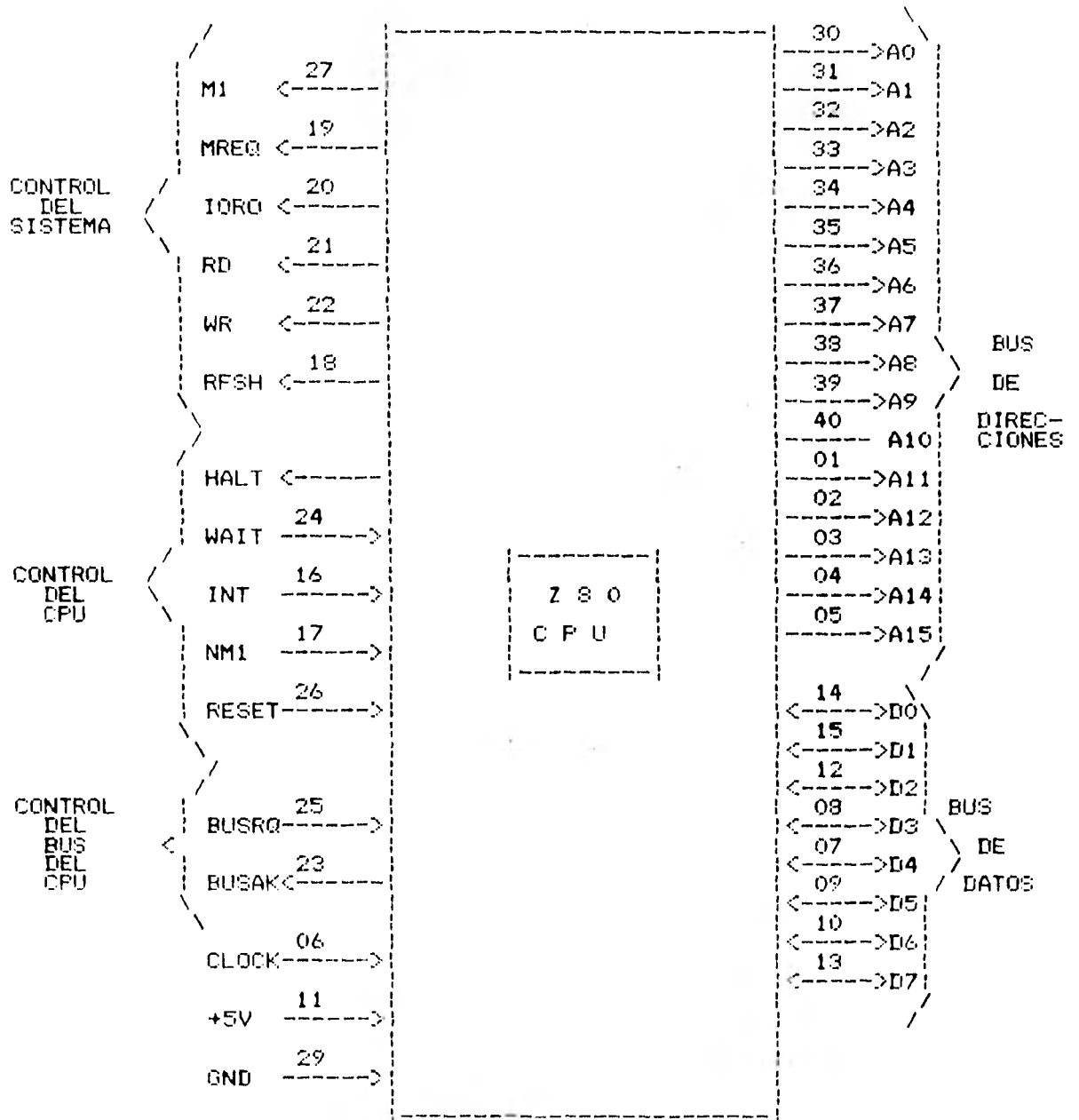
dan las direcciones para el acceso a este tipo de memoria.

- Conjunto de instrucciones del Z80-CPU.

El Z-80 CPU puede ejecutar 158 instrucciones diferentes; incluyendo las 78 instrucciones del 8080A CPU. Las instrucciones pueden ser desglosadas en los siguientes grupos:

- Carga e intercambio
- Transferencia de bloques y búsqueda
- Aritmética y Lógica
- Rotación y corrimiento
- Manipulación de bit (Set, Reset y Test)
- Salto, Llamada a rutinas y regreso de ellas
- Entrada/Salida
- Control básico del CPU

El CPU, físicamente hablando se encuentra "empacado" en una pastilla de plástico negra, de tal forma que cuenta con 40 "clavijas" ó pin. A continuación se muestra un diagrama de las mismas:



A continuación se explicará la función de cada una de las clavijas:

A0 - A15 BUS DE DIRECCIONES.

Ocupa de la clavija 0 a la 15 (A0 - A15). Es un bus de 16 bits. Provee la dirección de memoria (hasta 64K bits) para los intercambios de datos y para el intercambio de información con los dispositivos de Entrada/Salida.

D0 - D7 BUS DE DATOS.

Es un bus bidireccional de 8 bits. Es utilizado para intercambio de datos entre la memoria y los dispositivos de Entrada/Salida.

M1 (MACHINE CICLE ONE) CICLO DE MAQUINA UNO.

Esta señal indica que el ciclo actual de la máquina es el ciclo cargado del código OP de la ejecución de una instrucción.

MREQ MEMORY REQUEST.

Esta señal indica que el bus de direcciones contiene una dirección válida para una operación de memoria de lectura o escritura.

I0QR REQUEST DE ENTRADA, SALIDA.

Indica que la parte baja de la dirección del bus contiene una dirección válida de entrada o salida para una operación de lectura o escritura.

RD LECTURA DE MEMORIA.

Esta señal indica que el CPU quiere leer datos de memoria o de un dispositivo de E/S.

WR ESCRITURA DE MEMORIA.

Indica que el bus de datos del CPU contiene datos válidos para ser almacenados en el dispositivo de memoria direccionable o de E/S.

RFSH REFRESCAMIENTO (REFRESH).

Indica que los menores 7 bits de la dirección del bus contienen una dirección de refrescamiento para memorias dinámicas y que la señal actual MREQ debe ser utilizada para refrescar todas las memorias dinámicas.

HALT (HALT STATE) ESTADO DE ALTO.

Indica que el CPU ha ejecutado una instrucción de software de halt y está esperando una interrupción.

WAIT ESPERA.

Indica al CPU del Z-80 que la dirección de memoria o dispositivo de entrada o salida no está listo para una transferencia de datos.

INT (INTERRUPT REQUEST) SOLICITUD DE INTERRUPCION.

Esta señal es generada por los dispositivos de entrada y salida. Una solicitud de interrupción

será utilizada al final de la actual instrucción si el software interno controlado interrumpe el IFF (Flip-Flop O biestable) y si la señal BUSRQ no se encuentra activada.

NMI (NON MASKABLE INTERRUPT) INTERRUPCION NO ENMASCARABLE.

Esta instrucción fuerza automáticamente al CPU del Z-80 a empezar de nuevo en la localidad 0066H. El PC es salvado automáticamente en el stack externo, de tal forma que el usuario puede regresar al programa que fué interrumpido.

RESET.

Pone al PC en cero e inicia al CPU. Esto último implica lo siguiente:

- Inhabilita el Flip-Flop que estaba puesto en estado de interrupción.
- Coloca el registro I = 00H
- Pone al registro R = 00H
- Pone el modo de interrupción 0

BUSRQ BUS REQUEST.

Es utilizado para requerir la dirección del bus del CPU, del bus de datos y señales de control de tres estados para llegar a un estado de impedancia alta, de tal forma que otros dispositivos pueden controlar los buses.

BUSAK BUS ACKNOWLEDGE.

Es usado para indicar al dispositivo de requerimiento, que el bus de dirección es del CPU, el bus de datos y las señales de control del bus de tres estados han sido puestas a su estado de impedancia alta y el dispositivo externo puede ahora controlar estas señales.

* referencia 3.2

- Apéndice IV

Señales EIA RS232-C

Esta es una interfaz diseñada como estándar para tratar de asegurar que exista una compatibilidad entre los diferentes productos de diferentes fabricantes.

Las señales que se manejan se pueden dividir en:

- Circuitos de control.
- Circuitos de datos
- Circuitos de Temporización.

Esta interfaz físicamente tiene 25 pines distribuidos de la siguiente manera:

- 1.- Tierra de protección (FG).
Tipo: Tierra
- 2.- Transmisión de datos (TD).
Tipo: Datos
Dirección: Desde Equipo de Comunicación de Datos
- 3.- Recepción de datos (RD).
Tipo: Datos
Dirección: Desde Equipo de Comunicación de Datos
- 4.- Petición de transmisión (RTS).
Tipo: Control
Dirección: Hacia Equipo de Comunicación de Datos
- 5.- Preparado para transmitir (CTS)
Tipo: Control
Dirección: Desde Equipo de Comunicación de Datos
- 6.- Equipo de comunicación de datos listo (DSR).
Tipo: Control

- Dirección: Desde Equipo de Comunicación de Datos
- 7.- Tierra de señalización. (SG).
Tipo: Tierra
- 8.- Detector de Portadora (DCD)
Tipo: Control
Dirección: Desde Equipo de Comunicación de Datos
- 9.- Prueba del voltaje positivo del módem
- 10.- Prueba de voltaje negativo del módem
- 11.- Buffer saturado.
Tipo: Control
Dirección: Hacia Equipo de Comunicación de Datos
- 12.- Detector de Portadora de datos.
Tipo: Control
Dirección: Desde Equipo de Comunicación de Datos
- 13.- Preparado para transmitir secundario.
- 14.- Transmisión de datos secundario.
Tipo: Datos
Dirección: Desde Equipo de Comunicación de Datos
- 15.- Reloj de transmisión.
Tipo: Temporización.
Dirección: Hacia Equipo de Comunicación de Datos
- 16.- Recepción de datos secundario.
Tipo: Datos
Dirección: Desde Equipo de Comunicación de Datos
- 17.- Reloj de recepción.
Tipo: Temporización
Dirección: Desde Equipo de Comunicación de Datos
- 18.- Reloj de recepción de Díbits
Tipo: Temporización
Dirección: Desde Equipo de Comunicación de Datos
- 19.- Petición para transmitir secundario.
Tipo: Control
Dirección: Hacia Equipo de Comunicación de Datos
- 20.- Terminal de datos lista (DSR)

- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Control |
| | Dirección: | Hacia Equipo de Comunicación de Datos |
- 21.- Detector de calidad de señal.
- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Control |
| | Dirección: | Desde Equipo de Comunicación de Datos |
- 22.- Indicador de llamada (RI)
- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Control |
| | Dirección: | Desde Equipo de Comunicación de Datos |
- 23.- Selector de velocidad.
- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Control |
| | Dirección: | Hacia Equipo de Comunicación de Datos |
- 24.- Reloj externo de transmisión.
- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Temporización. |
| | Dirección: | Desde Equipo de Comunicación de Datos |
- 25.- Ocupado.
- | | | |
|--|------------|---------------------------------------|
| | Tipo: | Control |
| | Dirección: | Hacia Equipo de Comunicación de Datos |

* referencias 2.4 y 2.5

6.5.- Protocolo de comunicación de datos B-771.

6.5.1.- Formato del mensaje.

El formato del mensaje estará en código ASCII-67, y tendrá la siguiente configuración.

/ SOH / ARM / STX / DA1 / DA2 / <TEXTO> / ETX / BCC /

Las letras entre diagonales representan un solo carácter. El significado de ellos es el siguiente:

a).- SOH (Start Of Header).

Este es el carácter estándar ASCII (01H); el cual indica que se inicia la transmisión.

b).- ARM (Acknowledge Received Messase).

Este carácter es utilizado para indicar un reconocimiento de mensaje, ya sea positivo ó negativo, al mensaje previo, se forma de acuerdo a la paridad del mensaje y al número de la transmisión (TN).

Durante los mensajes iniciales, este carácter se inicia como el carácter NUL (00H). Cuando la secuencia (SOH-ARM) es recibida, esta es comparada con el último número de transmisión enviado. Si el ARM concuerda con el TN, el

Último mensaje enviado ha recibido acuse de recibo. Si el carácter no concuerda, el último mensaje debe ser retransmitido. Cuando un mensaje válido es recibido, el número de la transmisión contenido en ese mensaje es guardado para la utilización del siguiente carácter ARTM transmitido.

c).- TN (Transmission Number).

Este carácter es conocido como el número de la transmisión, este es la porción del mensaje que se utiliza para propósitos de acuse de recibo de mensajes, ya sean positivos ó negativos. Se utilizan solo dos números de transmisión, estos son: "f" y "u".

d).- STX (Start of Text).

Este es el carácter estándar ASCII de STX (02H), el cual sirve como delimitador entre la porción del encabezado precedente y la porción del texto del mensaje.

e).- DA1 y DA2 (Device Address)

Estas dos direcciones de dispositivos se utilizan para asociar un mensaje con uno de los dispositivos remotos en particular. Estos caracteres son utilizados por la computadora para enrutar datos al dispositivo remoto, y son insertados en los mensajes al ser transmitidos desde estos dispositivos.

f).- ETX (End of TeX)

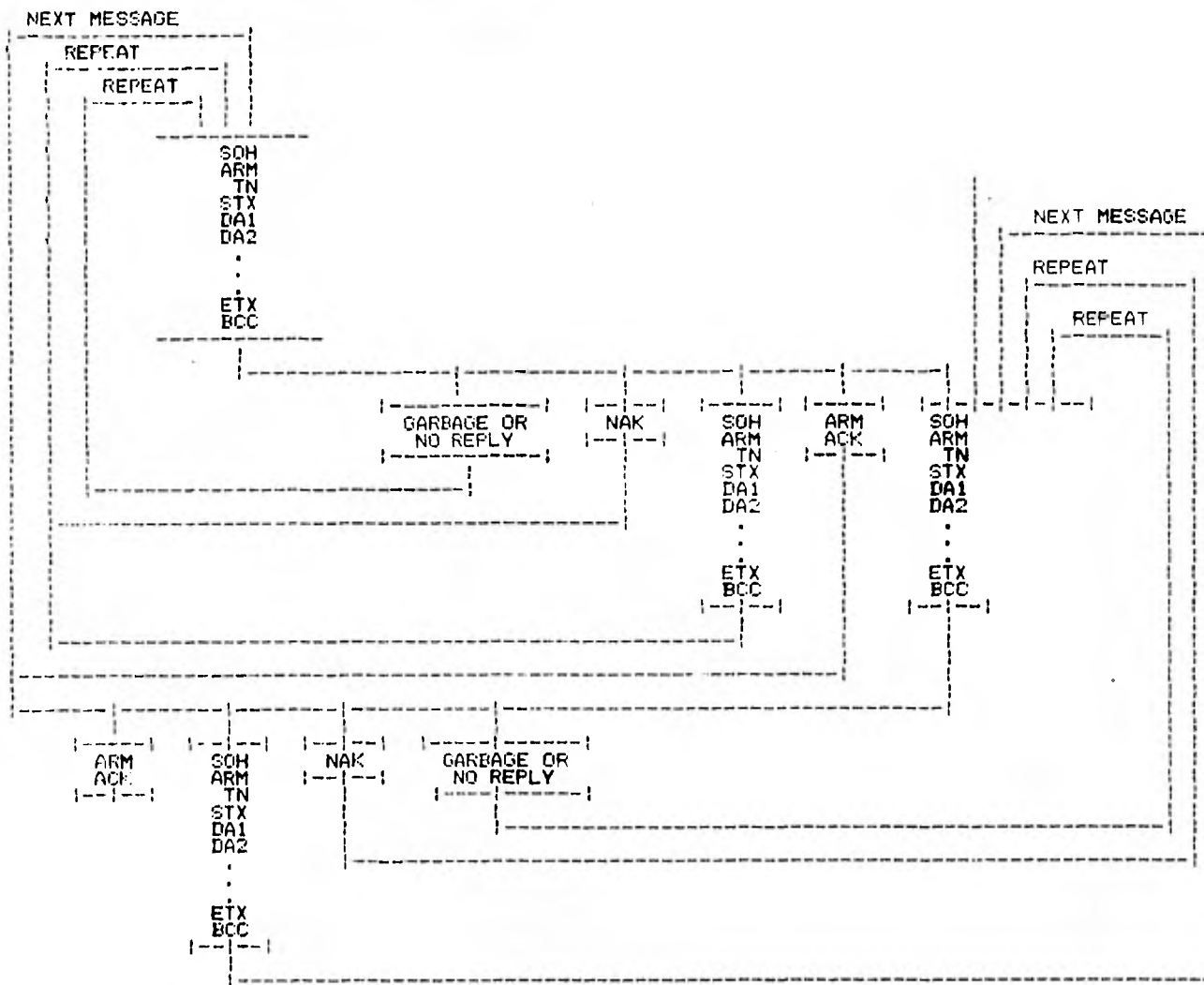
Este es el carácter estándar ASCII (03H); el cual se utiliza para indicar el fin del texto en un mensaje.

g).- BCC (Block Check Character)

Este es el carácter de la paridad longitudinal del mensaje. Este representa el "X-OR" de todos los caracteres que siguen al carácter "SOH" incluso hasta el carácter ETX.

El propósito de este carácter es el de asegurar la recepción correcta de los mensaje en el otro lado. Los mensajes conteniendo caracteres de BCC incorrectos recibirán acuse de recibo nesativo.

6.5.2.- Diagrama de comunicación.



* referencia 6.5

La presente tesis, fué
tecleada, editada e impresa
utilizando una microcomputadora
y los programas necesarios de
procesamiento de palabra.