4.4 Disk Tape Handler

The Disk/Tape Handler is an EPROM resident program to provide communication with the Technico disk or Technico tape. The two EPROM'S are located at address #F000 on the T99SS-CPU. The Disk/Tape Handler provides the following set of terminal commands:

AF	file length	allocate file
CD	file	catalog file
CF	file-1 file-2 length	copy file
DF	file	delete file
DU	file-no file	define unit
EX	address	execute
GD	/drive	gap drive
IB		inspect bound
ID	volume sequence	initialize disk
IR	file-no length	input record
IS	track sector	input sector
LF	file,address	load file(disk)
LT	file	load file(tape)
OR	file-no length	output record
OS	track sector	output sector
PF	file-no sector byte	position unit
RD		reset disk
RF	file	rename file
SB	bound-1,bound-2	set bounds
SF	file	save file(disk)
SO	date/drive	set options
ST	file	save file(tape)
TD	option/drive	test disk

4.4.1 Installing the EPROM'S

The Disk/Tape Handler is supplied in two 2708 EPROM'S. They are to be installed in locations U50, U51 of the CPU board. Be sure to locate pin one properly. Failure to do so will destroy the EPROM.

4.4.2 Interconnection/Operation of Technico Tape

The Technico Tape unit is connected to J1 and J2 of the basic CPU. An additional ground should be connected to any convenient ground. The signals, wires, pin locations and I/O bits are:

Signal	Wire	Connector	CRU Bit
/Request to Send	Yellow	J1-4	3-OUT
/Clear to Send	Red	J2-4	3-IN
Data Out (to CPU)	Orange	J2-3	2-IN
Data In (From CPU)	Green	J1-3	2-OUT
Ground	Black		
Unused	Blue-White		

Notice that /Request to Send and /Clear to Send are negative logic signals. That is, when /Request to Send is low (zero state) a request is made to the tape drive to send data. When high, no request is made and the tape unit is turned off. When the tape unit is off, the manual controls (play,rewind) are disabled. To enable these controls, the following monitor command will create a /Request to Send.

```
?M3 0
```

When recording data, it is recommended that the unit remain in PLAY/RECORD and that CRU bit 3 be used to stop the unit. Refer to the command description for Save Tape (ST) for further details.

4.4.3 Executing DTH

If the DTH EPROM'S are installed, the monitor will automatically transfer control to DTH after it has determined the terminal baud rate. To return to the monitor, enter the DTH command 'MON'. If the monitor is entered from DTH, control is returned to DTH by the monitor 'R' command. Before sending any commands to the disk, you must send a RD (reset) command to the disk. It is good practice to follow this with a SO (set options).

Whenever Disk/Tape Handler is entered, it will respond with a prompt character of greater than. You can now enter any Disk/Tape Handler command. Each of the commands is described in detail later. During entry of Disk/Tape commands, you can edit input as follows:

-BREAK. If you press BREAK the current line is discarded and a new line requested.

-RUBOUT or DEL. Each time a RUBOUT is received, the Disk/ Tape Handler will backspace one character. The character being deleted will be typed in response to RUBOUT.

4.4.4 DTH Commands

Many DTH commands require a file name for one parameter. A file name identifies a particular file or group of files. If only one file is specified by the name, it is unique. Certain commands (e.g. AF) require unique file names. If more than one file satisfies the name, it is ambiguous. Many commands allow ambiguous file names (e.g. CF). In that case all files which satisfy the file name criteria are utilized.

File names consist of four parts:

- Name - a sequence of up to eight characters
- Version(Optional) - a sequence of up to three characters seperated from the name by '.'.
- Type(Optional) - a single character seperated from the version (or name if no version) by a ':'.
- Drive (Optional) - a drive number (0 to 3) separated from the rest of the file name by '/' If omitted, the default drive is used. Refer to the Set Options (SO) command for further information regarding the default drive.

The general file name format is:

```
NNNNNNNN.VVV:T/D
```

The following are valid file names:

```
MON
MONITOR.SRC
EDITOR.REL:1
GEORGE:1/2
```

All of the above file names are unique since only one file could satisfy the name criteria. The special characters '?' and '*' may be used to create ambiguous names. The character '?' is a 'wild card' it will match any character. For example, the file name:

```
TECH???.WH?
```

will match all of the following file names:

```
TECHNIC.WHO
TECHX.WHX
TECHNI.WH
```

Note that '?' also matches a missing character. That is, "X??" matches "XY". The character '*' is used to denote that all remaining positions in the file name are '?'. The following examples show the correspondence between '?' and '*':

```
TEC.*    =   TEC.???:?
TE*      =   TE??????.???:?
*.REL    =   ????????.REL:?
*        =   ????????.???:?
```

All of the DTH commands are described in the following subparagraphs.

ALLOCATE FILE - Allocate an empty disk file of the specified length.

Format: AF file length

Procedure:

1. Type "AF"

2. Type the file name to be allocated followed by a space. If the diskette already contains a file with this name, an error message will be given.

3. Type the length in decimal number of sectors followed by a carriage return. The minimum allocation is one sector and the maximum is the number of sectors not currently allocated. Each sector contains 128 bytes of data.

Note:

1. The Editor/Assembler/Linking Loader (EAL) will automatically allocate files as 100 sectors. The AF command can be used to allocate longer files for use by EAL.

Example: The following command will allocate 130 sectors for file NEW.SRC on disk drive two:

```
AF NEW.SRC/2
```

CATALOG DISKETTE - Provides a catalog of the active files on the specified diskette.

Format: CD file

Procedure: 1. Type "CD"

2. Type the file name followed by a carriage return. The Disk/Tape Handler will now list the following information for the specified file(s):

o Name

o Start of allocation (decimal track/sector)

o Length of data (decimal sectors). This is the amount of the file currently in use.

o Date of creation-written on the disk only if the Set Options command was entered prior to file allocation.

o Date of last update-writen on the disk only if the Set Options command was entered prior to file update.

Note: 1. To obtain a catalog of all files on a given drive, use the file name */drive.

Example: The first command below will list all files on drive zero. The second will list only files with .SRC as part of the name.

```
CD */0
CD *.SRC
```

COPY FILE - Copy one file to another file.

Format: CF file-1 file-2 length

Procedure:

1. Type "CF"

2. Type the file(s) to be used as an input file.

3. Type the file(s) to be used as output. This file may have the same name or a different name than the input file.

4. Type the allocation of the output file (in decimal sectors) followed by carriage return.

Note:

1. The length parameter is optional and may be deleted. If deleted, file-2 is allocated the same length as file-1.

Example:

1. Copy all files from the default drive to drive 2.

```
CF * */2
```

The names and allocation are unchanged.

2. Copy file XYZ on the default drive to file ABC on the same drive, but change the allocation of ABC to 311 sectors.

```
CF XYZ ABC 311
```

3. Copy all file beginning with "XY" on the default drive to drive zero. Change the first two characters of the name of the file from "XY" to "AB".

```
CF XY* AB*/0
```

DEFINE UNIT- Associate a file number with a particular file.

Format: DU file-no file

Procedure:

1. Type "DU"

2. Type the number (1 to 5) to be used as the file-number.

3. Type the name of the file.

Note:

1. If the file name is omitted, the file associated with the file-no is closed.

2. If both the file-no and the file name are omitted, all files are closed.

Example: Open file XYZ.Y as file number 3.

```
DU 3 XYZ.Y
```

EXECUTE-Transfer control to a user program.

Format: EX Address

Procedure: 1. Type "EX"

2. Type the destination address. DTH will transfer control to that address.

Note: 1. The destination address can be omitted. In this case the first bound (see set bounds command) is used as a destination.

Example: Load file ABC at location 300 then execute that program.

```
LF ABC,300
EX
```

or;

```
SB,300
LF ABC
EX
```

GAP DISKETTE-compress the diskette to eliminate gaps between files.

Format: GD/drive

Procedure: 1. Type "GD"

2. Type the drive (e.g./2) followed by carriage return.

Note: 1. To gap the disk, the controller copies files from the end toward the beginning. If during any copy a hard error occurs, the file is lost. Therefore, it is highly recommended that a backup disk is made prior to any gap operation. A backup is made using the CF command.

2. The drive may be eliminated from the command. In this case, the default drive is utilized.

Example: If drive zero is the default, either of the following will gap drive zero:

```
GD/0
GD
```

INSPECT BOUNDS-Examine the bounds.

Format: IB

Procedure: 1. Type "IB"

2. The DTH will now type the memory bounds. Refer to set bounds command for a description of the bounds.

INITIALIZE DISK-Erase all files from the diskette, and format or initialize the diskette.

Format: ID volume sequence

Procedure:

1. Type "ID"

2. Type the volume identifier. A volume identifier is a file name used to label the disk. If /drive is part of the name (e.g.BACK.3/2) then that drive will be initialized.

3. Type the interleave sequence followed by a carriage return. Although the sequence can be from 1 to 13, a sequence of 4 will provide best performance with the TI9900.

4. The Disk/Tape Handler will now type "OK"? Check the command for errors. If you have not made an error type "Y", else type "N".

Note:

1. Initialize will delete all files currently stored on the diskette.

2. All new disks must be initialized before using them to store data.

Example: The following sequence will initialize the disk on drive zero and label it EXAM:

```
ID EXAM/0 4
OK?Y
```

INPUT RECORD-Input a record from a file currently open.

Format: IR file-no length

Procedure:

1. Type "IR"

2. Type the file-no of the file to be read. The file-no must be opened with a "DU" command.

3. For a binary record, type the length. For a variable length record, (like those for EAL) omit the length.

Example: Read a record on file 3 into 1000.

```
SB 1000
IR 3
```

INPUT SECTOR-Read a single sector of the diskette

Format: IS track sector

Procedure: 1. Type "IS"

2. Type the track (0 to 76) and sector (1 to 26) of the desired sector.

Note: 1. The data will be stored in 128 bytes of memory starting at bound-1 (see set bounds command)

Example: Read track 7 section 6 into locations 1000 to 107F

```
SB 1000 107F
IS 7 6
```

LOAD FILE-Load a disk file into memory.

Format: LF file, address

Procedure: 1. Type "LF"

2. Type the file name to be loaded followed by the target adress. The file name must be unique and must be seperated from the address by ",". DTH will now load the file at the specified address

Note: 1. The address of the memory to be used can be established via the Set Bounds (SB) command prior to loading a file. In this case the address is unnecessary.

Example: Load the Editor/Assembler/Linking Loader into RAM at address 300(hex).

```
SB 300
LF EAL
```

or;

```
LFEAL,300
```

LOAD TAPE-Load a tape file or record into memory.

Format: LT file

Procedure:

1. Type "LT"

2. Type the file name to be loaded followed by a carriage return.

3. Start the tape unit by depressing PLAY. The Disk/Tape Handler will skip to the desired file and load it.

4. When the Disk/Tape Handler prompt is issued, examine the bounds (via IB). If the second bound = FFFF, an error occurred during load.

Note:

1. Spaces in the file name are significant for tape files. Thus, the command "LTXYZ" and "LT XYZ" will not refer to the same file!

2. The file name may be deleted. In this case, the Disk/Tape Handler will load the next record on tape, regardless of identifier. This can be used to recover the name record of a file you have forgotton.

Example: Load the Editor/Assembler/Linking Loader into RAM at 300 (hex).

```
SB 300
LT EAL
```

Load the first record on tape (assume it has just been rewound. If that were the EAL tape, then the name "EAL" would be in memory at 300. That is, 300=20 45 41 4C. Using an ASCII conversion table, you could recover the file name.

```
SB 300
LT
```

OUTPUT RECORD-Output a record to a file currently open.

Format: OR file-no length

Procedure

1. Type "OR"

2. Type the file-no of the file to be written. The file-no must be opened with a "DU" command.

3. For a binary record (no record seperators), type the length. For a variable length ASCII record, omit the length.

Example: Write locations 1000 to 104F on file 2 as a binary record.

```
SB 1000 104F
OR 2
```

OUTPUT SECTOR-Write a single sector on the diskette.

Format: OS track sector

Procedure: 1. Type "OS"

2. Type the track and sector of the desired sector.

Note: 1. The bounds (see the set bound command) must define a 128 byte block of memory to be written on the diskette.

Example: Write locations 1000 to 107F on track 7 sector 6 of both drives zero and two.

```
SB 1000, 107F
OS 7 6
OS 7 6/2
```

POSITION FILE-Position any open file.

Format: PF file-no sector byte

Procedure:

1. Type "PF"

2. Type the file-no to be positioned. The file-no must be defined by a previous "DU" command.

3. Enter the sector and byte position required.

Note:

1. If the sector and byte are omitted, the DTH will print the current position of the file.

2. If the byte is omitted, byte zero is assumed.

Example: Position file-number three at the logical end.

```
PF 3 9999
```

RESET DISK-Reset the disk controller.

Format: RD

Procedure:

1. Type "RD" followed by a carriage return.

2. The controller will be reset.

Note:

1. Issue an RD command after initial power on.

2. RD will remove all options set by the Set Options (SO), and will close all open files.

RENAME FILE- Change the file name of one or more files.

Format: RF file-1 file-2

Procedure:

1. Type "RF"

2. Type the file(s) to be renamed followed by space.

3. Type the new name of the file(s) followed by carriage return.

Note:

1. The wild card characters "?" or "*" may be used as part of the file name (see example).

Example: Change all files of the form XXXXXXXX.SRC to XXXXXXXX.BAK.

```
RF *.SRC *.BAK
```

change the file EAL to EAL.ABS:1.

```
RF EAL EAL.ABS.1
```

SET BOUNDS-Change the memory bounds for load/store purposes.

Format: SB bound-1,bound-2

Procedure: 1. Type "SB"

2. Type the memory bounds in hexadecimal.

Note: 1. A Set Bounds command must be issued prior to a "SF" or "ST". This will define the area of memory to be saved.

2. A Set Bound command must be issued prior to a "LF" or "LT". In this case only the first bound is used. It is the starting address for the load. After the load, the bounds will be updated. A disk loak sets the bounds to the first and last adress read. A tape load sets the first bound to the last address +1 and the second will be set to the number of bytes just read or FFFF if an error occurred.

3. Bound-2 may be ommitted and the previous bound-2 will remain in effect.

Example: Set the bounds so that a program from disk or tape will be loaded at 1234 (hex).

SB 1234

Set bounds so that the area of memory from 1000 to 2163 (hex) can be stored on tape or disk.

SB 1000,2163

SAVE FILE-Store an area of memory on disk.

Format: SF file

Procedure:

1. Type "SF"

2. Type the file name to be assigned to the stored data. The file name must be unique and not already allocated.

3. The Disk/Tape Handler will allocate a file of the correct length and will store the desired information.

Note:

1. The address of the memory to be saved must be established via the Set Bounds (SB) command prior to storing a file.

Example: Save the Mighty Monitor memory on a file called MM.ABS. Data will be saved on drive two.

```
SB FC00,FFF
SF MM.ABS/2
```

SET OPTIONS-Establish the current date, and the default diskette drive.

Format: SO date /drive

Procedure:

1. Type "SO"

2. Type the date as a six character sequence. Any format can be used as long as it is six characters.

3. Type the default drive followed by a carriage return.

Note:

1. The option setting is reset whenever the disk controller is reset (see RD command).

2. The default drive can be omitted and zero is assumed.

EXample: Set the date to 1/3/78 and the default drive to zero.

SO 010378/0

or

SO 010378

SAVE TAPE-Write a tape file from memory

Format: ST file

Procedure: 1. Type "ST"

2. Type the name of the file to be assigned to the data. DO NOT type carriage return.

3. Start the tape unit by depressing PLAY/RECORD if not already set this way.

4. Type carriage return. After the data is written, stop the tape.

Note: 1. If you wish to record more than one file per tape, do not stop the tape after recording by pressing the stop switch on the tape. Instead, return to the monitor (via MON) and set bit 3 high (M3 1). The Disk/Tape Handler will restart the tape when another file is saved. If the tape is stopped manually by STOP, a glitch is recorded and causes read errors.

2. When writing the first record or file, be certain that you are past clear leader. This is done by setting bit 3 = low (M3 0) with the monitor, then pressing PLA/RECORD. When the leader is recorded, stop tape with the monitor (see Note 1).

Example: The following sequence of commands will save two files. File one is memory 1000 to 2000 and File two is 3100 to 3200. Monitor commands are those preceded by a "?". User entries underlined.

?M3 0 Allow rewind. At this point, insert cassette, rewind it, press RECORD/PLAY.

?M3 1 Stop tape (after passing clear leader)

?GF000 Go to Disk/Tape Handler

SB 1000,2000 Set up file-1

ST FILE1 Save file-1(space between ST FILE1 is significant)

MON Back to monitor when file output complete.

?M3 1 Stop tape

?R Back to Disk/Tape Handler

SB 3100,3200 Set up file-2

ST FILE2 Save file-2 When file output is complete, either stop tape using monitor (more files to come) or rewind and unload tape.

TEST DISK-Test Disk Drive.

Format: TD Option

Procedure:

1. Type "TD"

2. Type the option followed by a carriage return. The controller will test the disk according to the option as follows:

OPTION	TEST
V	Random seek/verify
R	Random seek/read
I	Incremental seek/read

3. To stop the test, reset the CPU, enter the Disk/Tape Handler and issue a Reset Disk (RD) Command.

Note: 1. If you have problems with a particular disk, try reinitializing it.

Example: Test drive two using a random seek/read sequence.

```
TD R/2
```

4.4.5 ERRORS

The Disk/Tape Handler will issue the following errors. If the error is encountered during execution of a terminal command, the message is printed. If the error is encountered during execution of a program request, the error is returned to the caller.

0007. Command is incorrect or a parameter improperly formed.

0006. An attempt to create a new file which duplicates a current file.

0005. The specified file was not found.

0004. The diskette is out of space. To regain space a gap command may help.

0003. The drive is not ready. This means it probably does not contain a diskette.

0002. An attempt was made to access a file via unit designator when no file is open. This may be caused by improper file definitions to the Editor, Assembler, Linker Loader. It is usually not encountered when using the utility commands of the Disk/Tape Handler.

0001. The current read or write cannot be completed because a hard error was encountered. This indicates a diskette problem-perhaps contamination by handling.

4.4.6 Program Listing

A complete listing of the EPROM resident software is included in this section. The following entry points are used by other system software (e.g. BASIC) and can be accessed by the user.

4.4.6.1 Disk Interface

A command is transferred to the disk by a call of the form:

```
LI R0,command address
LI R1,bound-1
L1 R2,bound-2
BLWP @#F804
DATA Error Address
```

If the error address is zero, all errors will be considered fatal and will cause DTH to return control to the monitor. If the error address is non-zero, control is transferred there in case of a disk error. R0 will contain the error number.

R0 must point to the command text. The command letter is determined by the following table:

COMMAND	LETTER USED
AF	A
CD	Q
CF	C
DF	D
DU	F
GD	G
ID	K
IR	R
IS	I
LF	L
OR	W
OS	O
PF	P
RD	X
RF	N
SF	S
TD	T

Any command not listed above cannot be sent to the disk. As an example, the following sequence will allocate a file XYZ as 100 sectors. Errors will be fatal. (R1, R2, not loaded since they are not use).

```
                LI          R0,TXT
                BLWP        @#F804
                DATA        0
                .
                .
                .

TXT             TEXT        'A XYZ 100'
                BYTE        >0D
```

4.4.6.2 Tape Interface

Two entry points are provided to control the tape unit. These are:

TTIN (BL @#F808)	Read Technico Tape. R1=memory to be loaded R15=file name pointer or zero R2=no. of characters(set by handler)
TTOUT (BL @#F80C)	Write Technico Tape R1=memory to save R2=no. of bytes.

```
                        TITL 'FLOPPY DISK/TECHNICO TAPE HANDLER 10/1/78'
0000            DTH     IDT
0000                    DREG
                        DEF  DISK,TTIN,TTOUT
0026            BREAK   EQU  >26              ; MONITOR BREAK AREA
FC4C            MONADR  EQU  >FC4C            ; RE-ENTRY TO MONITOR
0030            CRUBAS  EQU  >30              ; BASE=20, BUT R12 AT 8TH BIT
0004            TAPBAS  EQU  4                ; TAPE UNIT BASE
00B0            REG     EQU  >B0              ; DISK HANDLER REGISTERS
0080            MONWS   EQU  >80              ; DISK MONITOR REGISTERS
00D2            IBUFF   EQU  >D2              ; COMM. BUFFER(32 CHAR)
FFFC            SELECT  EQU  -4               ; OFFSETS FROM CRU BASE
FFFB            STATUS  EQU  -5               ; FOR THE FIVE CONTROL
FFFA            WRITE   EQU  -6               ; BITS.....
FFFD            NRESOT  EQU  -3
FFFB            NRESIN  EQU  -5
0004            EOT     EQU  >04              ; END OF TEXT
0001            SOH     EQU  >01              ; START OF HEADING
0005            ENQ     EQU  >05              ; ENQUIRY
0006            ACK     EQU  >06              ; ACKNOWLEDGE
0015            NAK     EQU  >15              ; NEGATIVE ACKNOWLEDGE
0000 0460 023E  BRNCH   B    @GO              ; MONITOR ENTRY
0004 00B0       DISK    DATA REG
0006 0010               DATA DISK1            ; SOFTWARE COMMUNICATION
0008 0460 03B8  TTIN    B    @LOAD
000C 0460 037E  TTOUT   B    @SAVE
                *
                * EQUIVALENCES FOR ERROR MESSAGES
                *
0007            CMDERR  EQU  >07              ; COMMAND ERROR
0010            INFORM  EQU  >10              ; INFORMATIONAL TEXT
0012            BADRES  EQU  >12              ; BAD RESPONSE TO OUR CMD
0021            SOHACK  EQU  >21              ; SOH, DATA, BUT NO ACK, EOT
0022            ENQDAT  EQU  >22              ; ENQ, EOT, ..., DATA
0023            ENQACK  EQU  >23              ; ENQ, EOT, ..., NOT ACK, EOT
                *
                * ENTRY POINT FOR DISK CONTROL
                *
0010 C0DD       DISK1   MOV  *R13,R3          ; GET USER'S R0 INTO R3
0012 C06D 0002          MOV  @2*R1(R13),R1    ; GET USER'S R1 INTO R1
0016 C0AD 0004          MOV  @2*R2(R13),R2    ; GET USER'S R2 INTO R2
001A C2BE               MOV  *R14+,R10        ; GET ERROR EXIT, ADJ. R14
001C 071D               SETO *R13             ; USER R0 = -1 = NO ERROR
001E 020C 0030          LI   R12,CRUBAS       ; INITIALIZE CRU BASE
0022 0209 00C4          LI   R9,ERROR1        ; SET FOR ERRORS
0026 1EFC               SBZ  SELECT           ; SELECT = 0
0028 1EFB               SBZ  STATUS           ; STATUS = 0
002A 1EFA               SBZ  WRITE            ; WRITE = 0
002C 9813 01D5          CB   *R3,@CHD         ; DELETE COMMAND?
0030 1316               JEQ  CONF             ; YES
0032 9813 01CF          CB   *R3,@CHK         ; KILL COMMAND?
0036 1313               JEQ  CONF             ; YES
0038 9813 04B2          CB   *R3,@X           ; X IS A RESET COMMAND
003C 161E               JNE  SEND             ; NOT RESET
                *
                * RESET THE DISK DRIVE
                *
003E 0209 0058          LI   R9,RESET3        ; SET FOR BAD RESET
```

```
0042 0204 F000  RESET  LI   R4,>F000          ; INIT. RESET COUNTER
0046 1EFD              SBZ  NRESOT
0048 0604       RESET1 DEC  R4
004A 16FE              JNE  RESET1
004C 1DFD              SBO  NRESOT
004E 1FFB       RESET2 TB   NRESIN
0050 16FE              JNE  RESET2           ; WAIT FOR RESET COMPLETE
0052 0203 01C4         LI   R3,CLSFIL        ; CLOSE ALL FILES
0056 1011              JMP  SEND
0058 0209 00C4  RESET3 LI   R9,ERROR1        ; TRY ONE MORE RESET
005C 10F2              JMP  RESET
                *
                * CONFIRM A KILL OR DELETE COMMAND WITH
                * THE OPERATOR.  HE MUST ANSWER 'Y' OR 'N'
                *
005E 020B 01CC  CONF   LI   R11,CMSS          ; SEND CONFIRM REQ.
0062 2C9B       CONF10 OUT  *R11
0064 058B              INC  R11
0066 D01B              MOVB *R11,R0
0068 16FC              JNE  CONF10
006A 2C40              IN   R0
006C 0980              SRL  R0,8
006E 0280 004E         CI   R0,'N'           ; NO?
0072 1325              JEQ  SEND50           ; THEN EXIT IMMEDIATELY
0074 0280 0059         CI   R0,'Y'           ; IF NOT 'Y', EXIT
0078 16F2              JNE  CONF
                *
                * SEND A COMMAND TO THE CONTROLLER
                * IF THE CONTROLLER HAS INFORMATION AVAILABLE
                * AT THIS TIME, FLUSH IT OUT SINCE IT IS
                * JUNK OR AN EOT FROM LAST COMMAND.
                *
007A 1DFB       SEND   SBO  STATUS
007C 1DFC              SBO  SELECT
007E 3600              STCR R0,8              ; READ STATUS
0080 0240 C000         ANDI R0,>C000          ; IS THERE DATA?
0084 1308              JEQ  SEND10            ; NO
0086 1EFB              SBZ  STATUS            ; YES-READ IT
0088 3600              STCR R0,8
008A 1EFC              SBZ  SELECT
008C 0200 0014         LI   R0,20             ; WASTE TIME FOR CONTROLLER
0090 0600       SEND5  DEC  R0
0092 16FE              JNE  SEND5
0094 10F2              JMP  SEND
0096 D033       SEND10 MOVB *R3+,R0           ; GET COMMAND BYTE
0098 06A0 01A0         BL   @DOUT             ; DISK OUTPUT
009C 0280 0D00         CI   R0,>0D00          ; CARRIAGE RETURN?
00A0 16FA              JNE  SEND10            ; NO-CONTINUE
                *
                * CARRIAGE RETURN FOUND
                *
00A2 06A0 0196         BL   @DOUTC            ; SEND AN EOT CHAR.
00A6 06A0 0180         BL   @DINP             ; DISK INPUT ROUTINE
00AA 1816              JOC  DCTRL             ; JMP IF A DISK CONTROL
00AC 06A0 0176         BL   @CRLF             ; SEND CARRIAGE RETURN
00B0 2C80       SEND30 OUT  R0                ; SEND DATA TO THE TERMINAL
00B2 06A0 0180         BL   @DINP             ; GET MORE FROM DISK
00B6 17FC              JNC  SEND30            ; NOT 'DCC' SO CONTINUE
```

```
00B8 0206 0010          LI   R6,INFORM          ; INFORMATIONAL TEXT
00BC C746         SEND40 MOV  R6,*R13            ; USER R0 = ERROR CODE
00BE 0380         SEND50 RTWP
                  *
                  * ROUTINE: ERRDAT (AND ERRDCC)
                  * READ 3 TYPES OF CHARACTERS FROM THE DISK AND SEND THEM
                  * TO TERM.  DATA CHARACTERS CONSIST OF THE PRINTABLE
                  * ASCII CHAR. AND THE ASCII CONTROL CHARACTERS (SUCH AS
                  * CARR. RET. (CR) AND END-OF-TEXT (EOT)).  DISK CONTROL
                  * CHARS. (DCC) CONSIST OF ASCII CONTROL CHARS. THAT
                  * WERE FLAGGED BY CONT. (VIA A NINTH I/O BIT) AS
                  * HAVING SPECIAL SIGNIFICANCE IN THE COMPUTER/CONTROLLER
                  * COMMUNICATION PROTOCOL.  DATA CHARACTERS ARE SENT TO THE
                  * TERM. AS IS, BUT DCC CHARACTERS ARE IGNORED.
                  *
00C0 04C0         ERRDCC CLR  R0                 ; CLEAR A 'DCC'
00C2 0459         ERRDAT B    *R9                ; PROCESS ERROR
00C4 06A0 0176    ERROR1 BL   @CRLF
00C8 2C80         ERROR2 OUT  R0
00CA 06A0 0180    ERROR3 BL   @DINP              ; GET ANOTHER DISK CHAR.
00CE 17FC              JNC  ERROR2               ; JMP IF NOT A 'DCC'
00D0 0280 0400         CI   R0,EOT*256           ; DID WE GET AN EOT YET?
00D4 1311              JEQ  ERRMSG               ; YES, TYPE AND ABORT.
00D6 10F9              JMP  ERROR3
                  *
                  * ROUTINE: DISK CONTROL
                  * ANALYZE A 'DCC' SEND IN RESPONSE TO COMMAND
                  *
00D8 0206 0012    DCTRL LI    R6,BADRES          ; BAD CONTROLLER RESPONSE
00DC 0280 0600         CI   R0,ACK*256
00E0 1336              JEQ  DRD60
00E2 0280 1500         CI   R0,NAK*256
00E6 1310              JEQ  NAK10
00E8 0280 0100         CI   R0,SOH*256           ; START OF HEADING
00EC 131B              JEQ  DREAD                ; DISK READ
00EE 0280 0500         CI   R0,ENQ*256           ; ENQUIRE?
00F2 132E              JEQ  DWRITE               ; DISK WRITE
00F4 C38A         TESTEX MOV  R10,R14            ; IS AN ERROR EXIT PROVIDED?
00F6 16E2              JNE  SEND40               ; YES-TAKE IT.
                  *
                  * THE CALLER HAS NOT PROVIDED ERROR EXIT, SO
                  * TYPE OUT R6 IN HEX AS THE ERROR CODE, RET. TO MONITOR
                  *
00F8 C146         ERRMSG MOV  R6,R5              ; SHOW ERROR NUMBER AS 4 HEX DI
00FA 1304              JEQ  ERRM10               ; NO ERROR CODE
00FC 06A0 0176         BL   @CRLF
0100 06A0 0218         BL   @TYPEWD              ; MONITOR ROUTINE
0104 0460 023E    ERRM10 B    @GO                ; EXIT TO DISK MONITOR
                  *
                  * ROUTINE: NAK
                  * A NEGATIVE ACKNOWLEDGE HAS BEEN RECEIVED.  ANALYZE IT.
                  *
0108 04C6         NAK10  CLR  R6                 ; IN CASE ERROR
010A C38A              MOV  R10,R14              ; ERROR EXIT PROVIDED?
010C 13D9              JEQ  ERRDCC               ; NO, SO SHOW THE NAK, ETC.
010E 06A0 0180         BL   @DINP
0112 0206 0007         LI   R6,CMDERR            ; COMMAND ERROR
0116 0204 01D4         LI   R4,TABLE             ; ADDRESS OF TABLE OF FIRST LET
```

```
011A 9D00          NAK20  CB    R0,*R4+          ; LOOK FOR A MATCH
011C 13CF                 JEQ   SEND40
011E 0606                 DEC   R6
0120 16FC                 JNE   NAK20
0122 10CC                 JMP   SEND40           ; BAD NAK
                   *
                   * ROUTINE: DISK READ
                   * SOH RECEIVED, SO READ FROM DISK.
                   *
0124 1DFB          DREAD  SBO   STATUS           ; SELECT STATUS
0126 1DFC                 SBO   SELECT
0128 1DFB          DRD10  SBO   STATUS           ; STATUS
012A 1F06          DRD20  TB    6                ; DATA AVAIL?
012C 1303                 JEQ   DRD30
012E 1F07                 TB    7                ; CONTROL AVAIL?
0130 1304                 JEQ   DRD40
0132 10FB                 JMP   DRD20            ; WAIT
0134 1EFB          DRD30  SBZ   STATUS           ; DATA
0136 3631                 STCR  *R1+,8           ; SAVE DATA
0138 10F7                 JMP   DRD10            ; CONTINUE
013A 06A0 018E     DRD40  BL    @DINP20          ; GET CONTROL CHAR
                   *
                   * DISK READ IS OVER WITH BECAUSE 'DCC' ENCOUNTERED
                   *
013E 0601                 DEC   R1
0140 CB41 0004            MOV   R1,@4(R13)       ; USER R2
0144 0206 0021            LI    R6,SOHACK        ; IF SOH, DATA, BUT NO ACK
0148 0280 0600     DRD50  CI    R0,ACK*256
014C 16B9                 JNE   ERRDCC           ; NOT ACK, SO IGNORE EXIT
014E 0380          DRD60  RTWP
                   *
                   * ROUTINE: DISK WRITE
                   * ENQ RECEIVED, SO WRITE TO THE DISK.
                   *
0150 06A0 0180     DWRITE BL    @DINP            ; FLUSH EOT
0154 D011          DWT10  MOVB  *R1,R0           ; GET THE DATA FOR DISK
0156 06A0 01A0            BL    @DOUT            ; SEND IT TO THE DISK
015A 0581                 INC   R1               ; MOVE THE BYTE POINTER
015C 1302                 JEQ   DWT20            ; R1 WENT TO 0000????
015E 8081                 C     R1,R2            ; CHECK IT AGAINST THE LIMIT
0160 12F9                 JLE   DWT10
0162 06A0 0196     DWT20  BL    @DOUTC
0166 0206 0022            LI    R6,ENQDAT        ; IN CASE ENQ, BUT NO EOT
016A 06A0 0180            BL    @DINP            ; DISK    IN
016E 17A9                 JNC   ERRDAT           ; NO 'DCC', SO IGNORE EXIT
0170 0206 0023            LI    R6,ENQACK        ; IN CASE ENQ, DATA, EOT,
0174 10E9                 JMP   DRD50            ; BUT NO ACK
                   *
                   * ROUTINE: CRLF
                   * SEND A CARRIAGE RETURN AND LINE FEED
                   * TO THE TERMINAL.
                   *
0176 2CA0 01CC     CRLF   OUT   @CR              ; SEND A CARRIAGE RETURN
017A 2CA0 01CD            OUT   @LF              ; SEND A LINE FEED
017E 045B                 B     *R11
                   *
                   * ROUTINE: DINP
                   * THIS ROUTINE INPUTS A BYTE FROM THE CONTROLLER
```

```
                    * AND SETS CARRY = 1 IF A CONTROL BYTE
                    *
0180 1DFB           DINP    SBO   STATUS            ; FIRST TEST STATUS
0182 1DFC                   SBO   SELECT
0184 3600           DINP10  STCR  R0,8              ; READ STATUS
0186 0240 C000              ANDI  R0,>C000          ; IS DISK SENDING TO US?
018A 13FC                   JEQ   DINP10            ; NO, READ STATUS AGAIN
018C 0A10                   SLA   R0,1              ; YES, SAVE H. O. IN CARRY
018E 1EFB           DINP20  SBZ   STATUS            ; PREPARE TO READ DATA
0190 3600                   STCR  R0,8              ; READ CHAR. OR CONTROL CHAR.
0192 1EFC                   SBZ   SELECT
0194 045B                   B     *R11
                    *
                    * ROUTINE: DOUT, DOUTC
                    * SEND A BYTE TO THE CONTROLLER
                    * ABORT IF CONTROLLER TRIES TO SEND
                    *
0196 0207 C000      DOUTC   LI    R7,>C000          ; R7=NOOP (MOV R0,R0)
019A 0200 0400              LI    R0,EOT*256        ; SEND EOT
019E 1002                   JMP   DOUT10
01A0 C1E0 018E      DOUT    MOV   @DINP20,R7        ; R7=SBZ STATUS
01A4 1DFB           DOUT10  SBO   STATUS            ; SELECT STATUS
01A6 1DFC                   SBO   SELECT
01A8 3604           DOUT20  STCR  R4,8              ; GET STATUS
01AA 0244 C300              ANDI  R4,>C300          ; READY?
01AE 1304                   JEQ   DOUT30            ; YES
01B0 0244 C000              ANDI  R4,>C000          ; IS CONTROLLER SENDING?
01B4 13F9                   JEQ   DOUT20            ; NO-WAIT
01B6 1085                   JMP   ERRDAT            ; YES-ERROR
01B8 0487           DOUT30  X     R7                ; SBZ STATUS IF REQUIRED
01BA 3200                   LDCR  R0,8              ; SEND
01BC 1DFA                   SBO   WRITE
01BE 1EFB                   SBZ   STATUS
01C0 1EFA                   SBZ   WRITE
01C2 045B                   B     *R11
                    *
                    * CONTROL DATA
                    *
01C4 46             CLSFIL  BYTE  'F'               ; CLOSE FILE
01C5 0D0A           EMSS    BYTE  >0D,>0A           ; MUST FOLLOW CLSFIL
01C7 3F3F 3F3F              TEXT  '???'             ; ERROR MESSAGE
01CB 00                     BYTE  0
01CC                CR      EQU   $
01CD                LF      EQU   $+1
01CC 0D0A           CMSS    BYTE  >0D,>0A
01CF                CHK     EQU   $+1
01CE 4F4B 3F                TEXT  'OK?'
01D1 00                     BYTE  0
01D2 3E             GT      BYTE  '>'               ; DISK MON. PROMPT
01D3 2C             COMMA   BYTE  ','
                    *
                    * TABLE OF CHAR. FOR NAK MESSAGES
                    *
01D4 43             TABLE   BYTE  'C'               ; 7 COMMAND ERROR
01D5 44             CHD     BYTE  'D'               ; 6 DUPLICATE FILE ERROR
01D6 4E                     BYTE  'N'               ; 5 NOT FOUND ERROR
01D7 4F                     BYTE  'O'               ; 4 OUT OF SPACE ERROR
01D8 52                     BYTE  'R'               ; 3 READY ERROR
```

```
01D9 55                 BYTE 'U'                ; 2 UNIT NUMBER ERROR
01DA 48                 BYTE 'H'                ; 1 HARD DISK ERROR
                        *                       ; 0 BAD CONTROLLER REPLY
                        ***********************************
                        * DISK MONITOR                     *
                        ***********************************
                        *
                        * ROUTINE: GETB
                        * GET A HEX VALUE FROM BUFFER
                        *
01DC 04C4               GETB    CLR  R4
01DE 04C5                       CLR  R5
01E0 D1B3               GETB10  MOVB *R3+,R6            ; GET NEXT CHAR
01E2 0986                       SRL  R6,8
01E4 0286 000D                  CI   R6,>0D             ; RETURN?
01E8 1315                       JEQ  GETB40             ; YES-EXIT
01EA 0226 FFD0                  AI   R6,->30
01EE 1110                       JLT  GETB30             ; NOT HEX
01F0 0286 000A                  CI   R6,10
01F4 1108                       JLT  GETB20             ; 0-9
01F6 0226 FFF9                  AI   R6,-7
01FA 0286 000A                  CI   R6,10
01FE 1108                       JLT  GETB30             ; NOT HEX
0200 0286 000F                  CI   R6,15
0204 1505                       JGT  GETB30             ; NOT HEX
0206 0205 0002          GETB20  LI   R5,2
020A 0A44                       SLA  R4,4
020C A106                       A    R6,R4
020E 10E8                       JMP  GETB10
0210 C145               GETB30  MOV  R5,R5              ; EXIT?
0212 13E6                       JEQ  GETB10             ; NO
0214 A2C5               GETB40  A    R5,R11             ; EXIT
0216 045B                       B    *R11
                        *
                        * ROUTINE: TYPEWD
                        * DISPLAY R5 AS HEX DIGIT STRING
                        * MONITOR ROUTINE NOT USED TO
                        * AVOID INCORRECT BREAK HANDLING
                        *
0218 0200 000C          TYPEWD LI   R0,12
021C C105               TYP10  MOV  R5,R4               ; GET 4 BITS
021E 0B04                      SRC  R4,R0
0220 0244 000F                 ANDI R4,>F
0224 0224 0030                 AI   R4,>30              ; CHANGE TO ASCII
0228 0284 003A                 CI   R4,>3A
022C 1102                      JLT  TYP20
022E 0224 0007                 AI   R4,7
0232 06C4               TYP20  SWPB R4
0234 2C84                      OUT  R4
0236 0220 FFFC                 AI   R0,-4               ; CONTINUE PER SHIFT COUNT
023A 18F0                      JOC  TYP10
023C 045B                      B    *R11                ; EXIT
                        *
                        * DISK MONITOR.  ACCEPT COMMANDS FROM THE
                        * TERMINAL AND SEND THEM TO DISK
                        * ALSO HANDLES TAPE UNIT
                        *
023E 02E0 0080          GO      LWPI MONWS              ; SET WORKSPACE
```

```
0242 C820 0000          MOV   @BRNCH,@4          ; SET LOC 4 TO RETURN
0246 0004
0248 C820 0002          MOV   @2,@6
024C 0006
024E 0200 023E          LI    R0,GO              ; SET BREAK HANDLER
0252 C800 0002          MOV   R0,@2
0256 06A0 0176          BL    @CRLF              ; PROMPT FOR COMMAND
025A 2CA0 01D2          OUT   @GT
025E 04C6               CLR   R6                 ; BUFFER LINE IN
                        *
                        * BUFFER ONE LINE FROM TERMINAL
                        * BREAK WILL ERASE THE LINE
                        *
0260 04C0       GO10    CLR   R0                 ; GET CHAR.
0262 2C40               IN    R0
0264 D980 00D2          MOVB  R0,@IBUFF(R6)      ; SAVE IT
0268 0280 7F00          CI    R0,>7F00           ; RUBOUT?
026C 130E               JEQ   GO30
026E 0586               INC   R6
0270 0280 0D00          CI    R0,>0D00           ; RETURN?
0274 16F5               JNE   GO10
                        *
                        * END OF THE INPUT LNE
                        *
0276 0203 00D2          LI    R3,IBUFF
027A 04C7               CLR   R7                 ; FIND IN TABLE
027C C027 046C  GO20    MOV   @CMD(R7),R0
0280 1309               JEQ   GO40               ; END OF TABLE
0282 8013               C     *R3,R0
0284 132A               JEQ   GO110              ; FOUND
0286 05C7               INCT  R7
0288 10F9               JMP   GO20
                        *
                        * HANDLE BACK SPACE
                        *
028A 0606       GO30    DEC   R6                 ; BACKSPACE
028C 11D8               JLT   GO                 ; TOO FAR
028E 2CA6 00D2          OUT   @IBUFF(R6)         ; ECHO CHAR.
0292 10E6               JMP   GO10
                        *
                        * CHECK FOR A SYSTEM FILE REFERENCE
                        * IF ROM BASED, JUMP TO IT
                        * IF RAM BASED, LOAD IT, THEN JUMP TO IT
                        *
0294 0207 04CA  GO40    LI    R7,PRGTB           ; LOOK UP NAME
0298 8DD3       GO50    C     *R3,*R7+
029A 1615               JNE   GO70
029C 8DE3 0002          C     @2(R3),*R7+
02A0 1613               JNE   GO80
02A2 C077               MOV   *R7+,R1            ; FOUND, GET ADDRESS
02A4 0B11               SRC   R1,1               ; PUT THE LSB
02A6 0A11               SLA   R1,1               ;    INTO THE CARRY
02A8 1708               JNC   GO60               ; IF EVEN, EXECUTE
02AA 0200 00D0          LI    R0,IBUFF-2         ; IF ODD, LOAD FIRST
02AE 0202 4C20          LI    R2,'L '
02B2 C402               MOV   R2,*R0
02B4 0420 0004          BLWP  @DISK              ; LOAD
02B8 0000               DATA  0
```

```
02BA C081        GO60    MOV  R1,R2              ; EXECUTE
02BC 0201 0026           LI   R1,BREAK           ; SET R1 FOR MONITOR
02C0 06A0 0176           BL   @CRLF              ; NEW LINE FIRST
02C4 0452                B    *R2
02C6 05C7        GO70    INCT R7                 ; NOT FOUND, TRY AGAIN
02C8 05C7        GO80    INCT R7
02CA C017                MOV  *R7,R0             ; END?
02CC 16E5                JNE  GO50
02CE 0207 01C5   GO90    LI   R7,EMSS            ; YES-ISSUE ERROR
02D2 D037        GO100   MOVB *R7+,R0
02D4 13B4                JEQ  GO
02D6 2C80                OUT  R0
02D8 10FC                JMP  GO100
                 *
                 * FOUND THE COMMAND LETTER IN THE TABLE,
                 * EXECUTE IT.
                 *
02DA C127 049C   GO110   MOV  @CMDS(R7),R4       ; DISK COMMAND?
02DE C004                MOV  R4,R0
02E0 0910                SRL  R0,1
02E2 1707                JNC  GO130              ; NO
02E4 0200 00D3   GO120   LI   R0,IBUFF+1
02E8 D404                MOVB R4,*R0             ; FIX COMMAND
                 *
                 * SEND COMMAND TO DISK
                 *
02EA 0420 0004           BLWP @DISK
02EE 0000                DATA 0
02F0 10A6                JMP  GO
                 *
                 * NOT DISK COMMAND
                 *
02F2 05C3        GO130   INCT R3
02F4 0454                B    *R4
                 *
                 * COMMAND: DISPLAY MEMORY BOUNDS (TAPE OR DISK)
                 *
02F6 06A0 0176   BOUND   BL   @CRLF              ; NEW LINE
02FA C141                MOV  R1,R5              ; DISPLAY BOUNDS
02FC 06A0 0218           BL   @TYPEWD
0300 2CA0 01D3           OUT  @COMMA
0304 C142                MOV  R2,R5
0306 06A0 0218           BL   @TYPEWD
030A 1099        SET10   JMP  GO
                 *
                 * COMMAND: SB X [,Y]
                 * SET MEMORY BOUNDS
                 *
030C 06A0 01DC   SETBD   BL   @GETB
0310 10DE                JMP  GO90               ; MUST HAVE ONE
0312 C044                MOV  R4,R1
0314 06A0 01DC           BL   @GETB
0318 1092                JMP  GO                 ; SECOND IS OPTIONAL
031A C084                MOV  R4,R2
031C 1090                JMP  GO
                 *
                 * COMMAND: LF <FILE> [,<ADDRESS>]
                 *
```

```
031E 0204 4C20  CMDLF   LI    R4,'L '
0322 9813 01D3  CMDLF1  CB    *R3,@COMMA        ; COMMA?
0326 1304               JEQ   CMDLF2
0328 9833 01CC          CB    *R3+,@CR
032C 16FA               JNE   CMDLF1
032E 10DA               JMP   GO120             ; JUST DO LOAD
0330 DCE0 01CC  CMDLF2  MOVB  @CR,*R3+
0334 06A0 01DC          BL    @GETB             ; SET BOUNDS
0338 1001               JMP   CMDLF3            ; NO ADDRESS
033A C044               MOV   R4,R1
033C 0204 4C20  CMDLF3  LI    R4,'L '           ; RESET R4
0340 10D1               JMP   GO120
                *
                * COMMAND: EX [<ADDRESS>]
                *
0342 06A0 01DC  CMDEX   BL    @GETB
0346 0451               B     *R1               ; NO ADDRESS
0348 0454               B     *R4               ; ADDRESS-USE IT
                *
                * COMMAND: TS [<FILE NAME>]
                * TAPE SAVE
                * SAVE R1 TO R2 ON TAPE
                * FILE ID IS IN IBUFF
                *
034A C201       TSV     MOV   R1,R8             ; SAVE R1,R2
034C C242               MOV   R2,R9
034E 0201 00D4          LI    R1,IBUFF+2        ; SAVE FILE NAME
0352 0226 FFFD          AI    R6,-3
0356 1303               JEQ   TSV10             ; NO NAME
0358 C086               MOV   R6,R2
035A 06A0 037E          BL    @SAVE             ; WRITE FILE NAME
035E C048       TSV10   MOV   R8,R1
0360 C089               MOV   R9,R2
0362 6081               S     R1,R2
0364 0582               INC   R2
0366 06A0 037E          BL    @SAVE             ; WRITE DATA
036A 10CF               JMP   SET10
                *
                * COMMAND: TL [<FILE NAME>]
                * TAPE LOAD - READ DATA INTO *R1
                * NAME OF THE REQUIRED FILE IS IN IBUFF
                *
036C 020F 00D4  TLD     LI    R15,IBUFF+2       ; POINT TO FILE NAME
0370 0226 FFFD          AI    R6,-3
0374 1601               JNE   TLD10
0376 04CF               CLR   R15               ; NO NAME
0378 06A0 03B8  TLD10   BL    @LOAD             ; LOAD THE FILE
037C 10C6               JMP   SET10
                * TAPE HANDLING ROUTINES
                * ----------------------
                *
                * ROUTINE: SAVE
                * WRITE THE CARACTERS IN MEMORY AT *R1 TO TAPE
                * R2=NO. OF CHARACTERS
                * FORMAT:
                * 7F 7F 7F 7F 7F 55 LEN(WORD) DATA
                *
037E C28B       SAVE    MOV   R11,R10           ; SAVE EXIT
```

```
0380 020C 0002          LI   R12,2
0384 1E00               SBZ  0                  ; DATA NORMALLY LOW
0386 1E01               SBZ  1                  ; RTS=LOW
0388 0203 0005          LI   R3,5               ; R3=NO OF 7F'S
038C 020F 0454          LI   R15,TWRITE         ; R15=INDEX TO WRITE
0390 1F01       SAV5    TB   1                  ; WAIT FOR CTS=LOW
0392 13FE               JEQ  SAV5
0394 0204 7FFF  SAV10   LI   R4,>7FFF           ; R4=CODE
0398 069F               BL   *R15
039A 0603               DEC  R3
039C 16FB               JNE  SAV10
039E 0204 5555          LI   R4,>5555
03A2 069F               BL   *R15
03A4 C102               MOV  R2,R4              ; WRITE LENGTH
03A6 069F               BL   *R15
03A8 C102               MOV  R2,R4
03AA 06C4               SWPB R4
03AC 069F               BL   *R15
03AE D131       SAV20   MOVB *R1+,R4
03B0 069F               BL   *R15               ;WRITE USER BYTE
03B2 0602               DEC  R2                 ; CONTINUE TILL END
03B4 16FC               JNE  SAV20
03B6 045A               B    *R10               ; EXIT
                *
                * ROUTINE: LOAD
                * READ RECORD INTO *R1 OR CHECK FILE NAME
                * RETURN THE NUMBER OF CHARACTERS IN R2
                * IF R15=0 THEN READ ELSE COMPARE FILE TO *R15
                *
03B8 C28B       LOAD    MOV  R11,R10
03BA 020C 0004  LD5     LI   R12,TAPBAS         ; SET CRU BASE
03BE 1E01               SBZ  1                  ; RTS=LOW
03C0 0208 042A          LI   R8,READ            ; INDEX TO READ
03C4 1F01       LD10    TB   1                  ; WAIT FOR CTS=LOW
03C6 13FE               JEQ  LD10
03C8 1F00       LD20    TB   0                  ; WAIT FOR DATA=LOW
03CA 13FE               JEQ  LD20
03CC 0209 FFF4          LI   R9,-12             ; R9=TIMER*8 LESS 76CYCLES
03D0 1F00       LD30    TB   0
03D2 16FE               JNE  LD30               ; WAIT FOR DATA=HIGH
03D4 1F00       LD35    TB   0                  ; SKIP ONE TRANSITION
03D6 13FE               JEQ  LD35               ; TO AVOID GLITCH
03D8 1F00       LD36    TB   0
03DA 16FE               JNE  LD36
03DC 0589       LD40    INC  R9                 ; TIME 8 BITS
03DE 1F00               TB   0
03E0 13FD               JEQ  LD40
03E2 0939               SRL  R9,3               ; R9=R9/8
03E4 0698       LD50    BL   *R8                ; READ AND SKIP >7F
03E6 0284 5500          CI   R4,>5500           ; 55?
03EA 1305               JEQ  LD60
03EC 0284 7F00          CI   R4,>7F00           ; >7F?
03F0 13F9               JEQ  LD50
03F2 0702               SETO R2                 ; ERROR
03F4 045A               B    *R10
03F6 0698       LD60    BL   *R8                ; GET BYTE LENGTH
03F8 D084               MOVB R4,R2
03FA 06C2               SWPB R2
```

```
03FC 0698                BL   *R8
03FE D084                MOVB R4,R2
0400 06C2                SWPB R2
0402 C0C2                MOV  R2,R3
                  *
                  * READ OR COMPARE FILE NAME
                  * R2=R3=RECORD LENGTH
                  *
0404 04C7                CLR  R7                 ; R7=LOCATE FLAG
0406 C20F                MOV  R15,R8             ; R8=INDEX OR FLAG
0408 06A0 042A   LD80    BL   @READ              ; GET BYTE
040C C208                MOV  R8,R8              ; STORE?
040E 1309                JEQ  LD100              ; YES
0410 9E04                CB   R4,*R8+            ; NO-COMPARE
0412 1301                JEQ  LD90
0414 0707                SETO R7                 ; SET NOT FOUND
0416 0603        LD90    DEC  R3                 ; CONTINUE TO END
0418 16F7                JNE  LD80
041A C1C7                MOV  R7,R7              ; IF R7=0, READ DATA
041C 16CE                JNE  LD5                ; ELSE SKIP FILE
041E 04CF                CLR  R15
0420 10CC                JMP  LD5
0422 DC44        LD100   MOVB R4,*R1+            ; SAVE DATA
0424 0603                DEC  R3                 ; CONTINUE TO END
0426 16F0                JNE  LD80
0428 045A                B    *R10               ; EXIT
                  *
                  * ROUTINE: READ
                  * READ ONE BYTE INTO R4
                  * R9=BAUD TIMER
                  *
042A 04C4        READ    CLR  R4
042C 0206 0009           LI   R6,9               ; R6=BIT COUNT
0430 1F00        RD5     TB   0                  ; WAIT TILL LOW
0432 13FE                JEQ  RD5
0434 1F00        RD10    TB   0                  ; WAIT FOR START
0436 16FE                JNE  RD10
0438 C149                MOV  R9,R5              ; WAIT 1/2 BIT
043A 1F00        RD20    TB   0
043C 0645                DECT R5
043E 15FD                JGT  RD20
0440 C149        RD30    MOV  R9,R5              ; (14) WAIT 1 BIT
0442 0914                SRL  R4,1               ; (14) REPOSITION
0444 1F00        RD40    TB   0
0446 0605                DEC  R5
0448 16FD                JNE  RD40
044A 3444                STCR R4,1               ; (42) SAMPLE
044C 0606                DEC  R6                 ; (10)
044E 16F8                JNE  RD30               ; (10) CONTINUE FOR 8
0450 06C4                SWPB R4                 ; DATA IN LEFT
0452 045B                B    *R11               ; EXIT
                  *
                  * ROUTINE: TWRITE
                  * WRITE THE BYTE IN R4 TO TAPE
                  *
0454 0205 000B   TWRITE  LI   R5,11              ; R5=BIT COUNT
0458 1D00                SBO  0                  ; START BIT
045A 0206 001F   WRT20   LI   R6,31              ; WAIT
```

```
045E 0606       WRT30  DEC  R6                ; DLY=20*R6+10 CYCLES
0460 16FE              JNE  WRT30
0462 3044              LDCR R4,1              ; SEND BIT
0464 0914              SRL  R4,1
0466 0605              DEC  R5                ; CONTINUE FOR ALL BITS
0468 16F8              JNE  WRT20
046A 045B              B    *R11              ; EXIT
                *
                * COMMAND TRANSLATION TABLE
                *
046C 4146       CMD    DATA 'AF'              ; ALLOCATE FILE
046E 4346              DATA 'CF'              ; COPY FILE
0470 4446              DATA 'DF'              ; DELETE FILE
0472 4744              DATA 'GD'              ; GAP DISK
0474 4944              DATA 'ID'              ; INITIALIZE(KILL) DISK
0476 4C46              DATA 'LF'              ; LOAD FILE
0478 534F              DATA 'SO'              ; SET OPTIONS(MODE)
047A 5246              DATA 'RF'              ; RENAME(NAME) FILE
047C 4344              DATA 'CD'              ; CATALOG DISK
047E 5346              DATA 'SF'              ; SAVE FILE
0480 5444              DATA 'TD'              ; TEST DISK
0482 5244              DATA 'RD'              ; RESET CONTROLLER(X)
0484 4952              DATA 'IR'              ; INPUT RECORD
0486 4953              DATA 'IS'              ; INPUT SECTOR
0488 4F52              DATA 'OR'              ; OUTPUT RECORD
048A 4F53              DATA 'OS'              ; OUTPUT SECTOR
048C 5046              DATA 'PF'              ; POSITION FILE
048E 4455              DATA 'DU'              ; DEFINE UNIT
0490 5342              DATA 'SB'              ; SET BOUNDS
0492 4C54              DATA 'LT'              ; LOAD TAPE FILE
0494 5354              DATA 'ST'              ; SAVE TAPE FILE
0496 4942              DATA 'IB'              ; INSPECT BOUNDS
0498 4558              DATA 'EX'              ; EXECUTE
049A 0000              DATA 0                 ; END OF TABLE
049C 4101       CMDS   BYTE 'A',1
049E 4301              BYTE 'C',1
04A0 4401              BYTE 'D',1
04A2 4701              BYTE 'G',1
04A4 4B01              BYTE 'K',1
04A6 031E              DATA CMDLF
04A8 4D01              BYTE 'M',1
04AA 4E01              BYTE 'N',1
04AC 5101              BYTE 'Q',1
04AE 5301              BYTE 'S',1
04B0 5401              BYTE 'T',1
04B2 5801       X      BYTE 'X',1
04B4 5201              BYTE 'R',1
04B6 4901              BYTE 'I',1
04B8 5701              BYTE 'W',1
04BA 4F01              BYTE 'O',1
04BC 5001              BYTE 'P',1
04BE 4601              BYTE 'F',1
04C0 030C              DATA SETBD
04C2 036C              DATA TLD
04C4 034A              DATA TSV
04C6 02F6              DATA BOUND
04C8 0342              DATA CMDEX
                *
```

```
                    * TABLE OF SYSTEM ROUTINES
                    * EACH ENTRY IS OF THE FORM:
                    * 4 BYTES - NAME OF ROUTINE
                    * 2 BYTES - ADDRESS
                    *
                    * IF ADDRESS IS ODD, FILE IS LOADED
                    * IF ADDRESS IS EVEN, MONITOR JUST BRANCHES
                    *
04CA 4D4F 4E        PRGTB  TEXT 'MON'           ; MONITOR
04CD 0D                    BYTE >0D
04CE FC4C                  DATA MONADR          ; IS AT MONADR
04D0 4949 41               TEXT 'IIA'           ; IIA
04D3 0D                    BYTE >0D
04D4 F800                  DATA >F800           ; IS AT >F800
04D6 4241 53               TEXT 'BAS'           ; BASIC
04D9 0D                    BYTE >0D
04DA 0101                  DATA >100+1          ; IS LOADED AT >100
04DC 4541 4C               TEXT 'EAL'           ; EAL
04DF 0D                    BYTE >0D
04E0 0301                  DATA >300+1          ; IS LOADED AT >300
04E2 464F 52               TEXT 'FOR'           ; FORTRAN
04E5 0D                    BYTE >0D
04E6 0101                  DATA >101            ; IS LOADED AT >100
04E8 5245 54               TEXT 'RET'           ; RETURN
04EB 0D                    BYTE >0D
04EC 0004                  DATA >0004           ; TO CALLING ROUTINE
04EE                       END
```

```
0006 ACK      0012 BADRES   02F6 BOUND    0026 BREAK    0000 BRNCH
01D5 CHD      01CF CHK      01C4 CLSFIL   046C CMD      0007 CMDERR
0342 CMDEX    031E CMDLF    0322 CMDLF1   0330 CMDLF2   033C CMDLF3
049C CMDS     01CC CMSS     01D3 COMMA    005E CONF     0062 CONF10
01CC CR       0176 CRLF     0030 CRUBAS   00D8 DCTRL    0180 DINP
0184 DINP10   018E DINP20   0004 DISK     0010 DISK1    01A0 DOUT
01A4 DOUT10   01A8 DOUT20   01B8 DOUT30   0196 DOUTC    0128 DRD10
012A DRD20    0134 DRD30    013A DRD40    0148 DRD50    014E DRD60
0124 DREAD   *0000 DTH      0150 DWRITE   0154 DWT10    0162 DWT20
01C5 EMSS     0005 ENQ      0023 ENQACK   0022 ENQDAT   0004 EOT
00C2 ERRDAT   00C0 ERRDCC   0104 ERRM10   00F8 ERRMSG   00C4 ERROR1
00C8 ERROR2   00CA ERROR3   01DC GETB     01E0 GETB10   0206 GETB20
0210 GETB30   0214 GETB40   023E GO       0260 GO10     02D2 GO100
02DA GO110    02E4 GO120    02F2 GO130    027C GO20     028A GO30
0294 GO40     0298 GO50     02BA GO60     02C6 GO70     02C8 GO80
02CE GO90     01D2 GT       00D2 IBUFF    0010 INFORM   03C4 LD10
0422 LD100    03C8 LD20     03D0 LD30     03D4 LD35     03D8 LD36
03DC LD40     03BA LD5      03E4 LD50     03F6 LD60     0408 LD80
0416 LD90     01CD LF       03B8 LOAD     FC4C MONADR   0080 MONWS
0015 NAK      0108 NAK10    011A NAK20    FFFB NRESIN   FFFD NRESOT
04CA PRGTB    0000 R0       0001 R1       000A R10      000B R11
000C R12      000D R13      000E R14      000F R15      0002 R2
0003 R3       0004 R4       0005 R5       0006 R6       0007 R7
0008 R8       0009 R9       0434 RD10     043A RD20     0440 RD30
0444 RD40     0430 RD5      042A READ     00B0 REG      0042 RESET
0048 RESET1   004E RESET2   0058 RESET3   0394 SAV10    03AE SAV20
0390 SAV5     037E SAVE     FFFC SELECT   007A SEND     0096 SEND10
```

```
 00B0 SEND30   00BC SEND40   0090 SEND5    00BE SEND50   030A SET10
 030C SETBD    0001 SOH      0021 SOHACK   FFFB STATUS   01D4 TABLE
 )04 TAPBAS  *00F4 TESTEX   036C TLD      0378 TLD10    034A TSV
 035E TSV10  *0008 TTIN    *000C TTOUT    0454 TWRITE   021C TYP10
 0232 TYP20    0218 TYPEWD   FFFA WRITE   045A WRT20    045E WRT30
 04B2 X
EDIT/ASM/LOAD?
```