PRODUCT CODE:	DEC-12-UR1A-D
PRODUCT NAME:	AIPOS Monitor Internal Description
DATE CREATED:	May 5, 1971
MAINTAINER:	Software Services

May, 1971

The material in this manual is for information purposes and is subject to change without notice.

For additional copies, order No. DEC-12-UR1A-D from Program Library, Digital Equipment Corporation, Maynard, Massachusetts 01754 Price $5.00

CONTENTS

CONTENTS

MONITOR INTERNAL INFORMATION

1.0 HOW TO USE THE MONITOR

1.1 Data File Format

AIPOS data files have two parts: HEADER and DATA.

The HEADER portion is composed of one primary header block and several optional secondary header blocks.

The primary header block contains information about the number of header blocks, the number of data blocks, and the data file in general. Its format is fixed as follows:

Loc Ø	Number of header blocks
Loc 1	Number of data blocks
Loc 2	Data format code Ø1Ø1 1 word integers Ø2Ø2 2 word integers Ø2Ø3 2 word fractions Ø3Ø4 3 word floating point
Loc 3- Loc 4	Number of data points in the file (2 word integer)
Loc 5- Loc 17	Reserved
Loc 20- Loc 22	Maximum value of the data file in 3 word floating point format
Loc 24- Loc 26	Minimum value of the data file in 3 word floating point format
Loc 30- Loc 377	Reserved

The secondary header blocks contain information on particular programs in AIPOS.

Loc Ø of all secondary header blocks contains a code defining the particular systems program. Currently, codes 1, 2, and 3 are assigned.

It is the responsibility of the programs that generated the secondary header blocks to set the format of the remaining locations of the secondary header blocks.

The block in the DATA portion of AIPOS data files contain the data points whose format is defined in the primary header block.

The number of data points per data block varies with the data format:

1 word integers	256_{10} points per block
2 word integers	128_{10} points per block
2 word fractions	128_{10} points per block
3 word floating point	85_{10} points per block

The last location of each floating point data block is unused.

1.2 Monitor Organization

The AIPOS core-resident Monitor provides device-independent I/O services and arbitrary, device-specific interrupt handling for all programs in the system. Because the system is structured in this way, each application program can be written with minimum concern for the details of device manipulation and control; yet where necessary for special needs or problems, the program has direct access to the hardware. In general, therefore, each program is free of the need for special I/O routines (and consequently the need to debug each routine), while the availability of a new peripheral on the system improves the power of every program, without recoding each one. Clearly, the software is as easily expandable as the hardware, and the user who wishes to add to his machine configuration need not worry about rewriting his programs to take advantage of it.

Similarly, file handling and command interpretation are handled by the Job Control Language processor at the beginning and end of every program run. The program itself need never be concerned with index structure or generalized command decoding, since these problems are handled by JCL, which converts the appropriate command and file information to a convenient table form for the application program.

The following information on the Monitor is included in this manual for the programmer who wishes to add a program to AIPOS. It is suggested that the AIPOS Internal Descriptions also be read.

The basic resident AIPOS Monitor occupies locations 2ØØ-2577 of field Ø. Page Ø is allocated as follows:

Ø-1	8-mode interrupt	
2-7	reserved for Monitor[1]	
1Ø-15	available for user program	
16	interrupt stack pointer	
17	Monitor call stack pointer	
2Ø-37	addresses of user-callable Monitor routines	
2Ø	SETINT	enable interrupt handler
21	READ	
22	WRITE	
23	WAIT	
24	MWAIT[2]	multiple WAIT
25	EXIT	return to Job Control
26	ABORT[2]	
27	INTPSH	
3Ø	INTPOP	
4Ø-43	LINCmode interrupt	
44-137	available for user program. Loaded by loader	
14Ø-177	reserved for Monitor use	

Optional Monitor facilities, available via conditional assembly in the Monitor, will increase the highest address used by the Monitor. These facilities include support of optional I/O devices, plus a grid table and character display routine for ASCII or DIAL codes.

The loader, when requested, loads locations 44-137 and 26ØØ-7577 of field Ø, and any location in field 1.

Page 37 of field Ø (locations 76ØØ-7777) serves as a communication region between Job Control and the user program. The information stored

[1] These six locations, tagged TMP1, TMP2,...,TMP6, are used by the Monitor as temporaries during user Monitor calls, but never in interrupt processing. The user may, therefore, use these locations as temporaries, with the knowledge that they may be destroyed by any Monitor call.

[2] Reserved, but not yet implemented.

here is file information, parameter strings, etc., and is passed in 12_8 word blocks having the following format:

Ø	unit number
1	block length (1 if not tape or disk)
2-4	file name (6 characters stripped ASCII)
5-6	extension
7	auxiliary (taken from index entry)
1Ø	starting block of file
11	length in blocks

Bits Ø and 1 of the unit word (word Ø) have the following meanings:

Bit Ø: Permanent/Temporary file. When the application program is called, this bit is zero if Job Control allocated new file space for this file, one if the file existed prior to loading the program. The bit will always be set for input files.

Bit 1: Input/Output. This bit is zero for files specified as input in the command string (i.e., specified to the right of the equals sign), and one for output files (including working areas).

Upon normal EXIT, Job Control will scan the descriptor blocks, making permanent index entries for those files whose descriptor blocks have both bits Ø and 1 set.

Words 2-11 are copied directly from the index of the specified device, if tape or disk. Otherwise, 2-6 contain blanks, 7 and 1Ø contain zeros, and 11 contains 7777.

These file parameter blocks are in page 37 as follows:

76ØØ	Command input file descriptor
7612	Load file descriptor
7624	First output file, or first working area descriptor if no current files
7636	Second output file or working area descriptor
765Ø	Third output file or first input file descriptor
7662	I/O file descriptor
7674	I/O file descriptor
77Ø6	I/O file descriptor
772Ø	I/O file descriptor
7732	I/O file descriptor
7744	Highest valid CDF for this machine
7745-47	Unassigned
775Ø-55	Various data break locations
7756	Unassigned
7757	Two's complement of parameter string length
776Ø-77	Parameter string in DIAL code (packed 6 bit)

A convention has been established for use of the Monitor EXIT routine: Normal exit from any application program is a JMP instruction, whereas error exits are a JMS. This distinction allows the Monitor to suppress saving of files when an application program is aborted, either by CTRL/C or under program control.

1.3 I/O Facilities

The standard calling sequence for I/O routines is as follows:

```
    CLA                         /not necessary if AC is clear
    RIF                         /not necessary if in field Ø
    CIF       Ø                 /not necessary if in field Ø
    JMS I  READ (or WRITE)
    PARAM                       /pointer to parameter list in this field
```

Immediately following the JMS is the address of a parameter list in the same field, whose format is:

```
   PARAM,   UNIT                   /I/O unit code (see table below)
            CDF    Y               /field containing buffer
            BUFFER                 /address of buffer in field
            BUFLEN                 /length of buffer in words (Ø
                                   /implies 4Ø96$_{1Ø}$)
            BLOCK                  /starting blk of transfer if
                                   /tape or disk.  Ignored on sequential
                                   /devices
            Ø                      /two words used for
            Ø                      /queueing requests
```

When the program jumps to READ or WRITE, the Monitor sets bit Ø of the unit code, and may change bits 1 through 5. The Monitor also destroys words 5 and 6 of the list in the process of queueing the request. The rest of the parameter list is not modified by the Monitor and must not be modified by the user until bit Ø of word zero is cleared. (This is the Monitor's indication that the operation is complete.) When the Monitor returns from the JMS, the link is set, the AC is zero, and the instruction and data fields are the same as they were before entering the calling sequence. All data transfers are limited to one field, that specified by the CDF instruction in the parameter list.

The unit code assignments are:

Mnemonic	Code	Meaning
	Ø	Illegal
	1-17	Reserved for later implementation of logical units
LTØ-7	2Ø-27	LINCtape units Ø-7
DKØ-3	3Ø-33	RK8 disk drivers Ø-3
TTY	4Ø	Console teleprinter
TTY1-TTY7[1]	41-47	Remote teleprinters
MTØ-3[1]	5Ø-53	MAGtapes
PTR, PTP[1]	54	High speed paper tape reader and punch
CDR, CDP[1]	55	Card reader and punch
LPT[1]	56	Line printer
	57	Unassigned
DSP, DSA	6Ø	ASCII character display
DSD	61	DIAL code character display
	62-67	Reserved for plotter and other displays
	7Ø-77	Unassigned

Internally, all references to I/O units are via the above codes.

Externally, all references are by the mnemonics.

Despite the common unit code for teleprinter input and output, the keyboard and printer are handled as logically distinct devices, and operations on them may be in progress concurrently. If the user program requests both input and output on the teleprinter, his output request will be honored immediately, but will be interrupted to echo anything typed by the operator. If no input has been requested, on the other hand, anything typed by the operator (except CTRL/C) is ignored, and output messages are not affected.

The user program may test for completion of any operation by examining bit Ø of word Ø of his parameter list, as follows:

```
          TAD PARAM          /get status
          SPA CLA            /skip if operation complete
          JMP POTDONE        /not done yet
```

Alternatively, a Monitor call is provided for the same purpose. The calling sequence is identical to that for I/O operations.

[1] Reserved, but not yet implemented.

```
        CLA                     /not necessary if AC is zero
        RIF                     /unnecessary in field Ø
        CIF     Ø               /unnecessary in field Ø
        JMS I   WAIT            /wait for completion
        PARAM                   /of this operation
```

The Monitor returns when bit Ø of the specified parameter list is zero, i.e., the operation is complete. (Note that ALT MODE is stored in the user's buffer as octal 233 whenever any of the characters 233, 375, or 376 is typed on the keyboard. The parity bit (2ØØ) is forced on before any keyboard character is stored in the user's buffer.

1.4 Interrupt Facilities

The Interrupt Handler is set up as follows:

```
        CLA                     /
        RIF                     /
        CIF     Ø               /Not necessary if IF already = Ø
        JMS I   SETINT          /Enable interrupt handler
        IOT                     /Skip on desired flag
        HANDLER                 /Address of interrupt handler
                                /in this field
```

The location following the JMS is assumed to be an IOT skip instruction. The next location is the address of an interrupt handler for the particular flag tested. The handler must be in the same field as the initialization call. When an interrupt occurs, and the tested flag causes a skip, the Monitor will JMS to the specified handler. AC and LINC may be anything, data and instruction fields will be the user's field (i.e., the instruction field when the SETINT was done). If the MQ is used, it must be saved and restored. If LINC mode adds are used, the FLO flip-flop must be saved and restored.

If the handler address is 7777, the specified skip instruction is removed from the polling sequence, thereby effectively disabling the specified interrupt handler. The maximum number of interrupts which may be enabled at any one time is specified at system assembly time, and includes interrupts from I/O devices in operation due to read/write calls not yet satisfied.

For each enabled interrupt (that is, each one for which a SETINT is active) the Monitor sets up the following instruction sequence:

```
IOT             /Skip taken from SETINT call
JMP .+3         /Test next if flag not set
CIF CDF X       /Set to user's field
JMS I USER      /Call user's routine
```

Disabling an interrupt overlays the IOT with a NOP (7ØØØ). When SETINT is called to enable an interrupt, the chain is scanned first to find an identical IOT. If a match is found, it is NOPed. Then the chain is scanned again, this time for a NOP. When one is found, the CIF CDF is replaced to address the user's field, the appropriate entry in the list of user routine addresses is set, and finally, the NOP is replaced by the IOT skip.

Each time an interrupt occurs, the entire chain is scanned once to ensure that all flags can be serviced with a minimum of overhead. It is the responsibility of the user to clear flags before returning. Failure to do so will result in an infinite interrupt loop.

It is also the user's responsibility to ensure that the time taken by his interrupt routine is small.

Two Monitor routines are provided to allow a user's interrupt routine to be interruptable. These routines are INTPSH and INTPOP. They save and restore, respectively, the system status at the time of the interrupt so that other interrupts may be serviced as necessary during the execution of the user's interrupt routine. Calling sequences are:

```
CLA
RIF
CIF    Ø
JMS I  INTPSH
ION
.                      /interruptable portion of user's
.                      /interrupt service routine
.
.
CLA
RIF
CIF    Ø
IOF
JMS I  INTPOP
```

Note that these routines provide for saving of system status, but not the status of the user's interrupt routine. Therefore, the user must protect against being re-entered by another interrupt from the same device. At the minimum, this means that the device flag must be cleared before calling INTPSH.

The programmer must make a clear distinction between interrupt service routines (i.e., synchronous or real-time routines) and background programs. At interrupt time, only two Monitor routines may be called: INTPSH and INTPOP. Since this rule can not be enforced by the Monitor, the programmer must use care not to violate it, and is warned that violations will be punished by pseudo-random, intraceable, and unreproducable system crashes.

1.5 Control Characters

Control characters (that is, those characters whose ASCII codes are between 2ØØ and 237 octal, inclusive) may optionally be trapped to a user program, to be handled at his discretion. To use this facility, the user program must store in location 177 (symbolic TTPCTL) of field Ø, the address of a subroutine to handle control characters. From this time until EXIT, CTRL/C, or restoration of location 177, all control characters typed on the console TTY[1] will be passed to the user subroutine. The subroutine may ignore the character, in which case it will be handled as before or it may request that the character be echoed as up-arrow followed by the character typed with control, or it may request that the character be completely ignored, neither stored in the buffer nor echoed. The subroutine may even change the character, in which case the new character will be treated as though it had been typed.

The user subroutine is called at interrupt time, whether or not a read is currently active on the TTY, after clearing the flag and turning interrupts on, by the following instruction sequence:

[1] TTY is an abbreviation for Teletype, a registered trademark of Teletype Corporation

```
                  JMS I        177
                  CHAR            /THE CONTROL CHAR IS HERE
NORMAL RETURN:    JMP          NRM/IF INPUT ACTIVE, ECHO CHAR AND
                                  /STORE IN USER'S BUFFER

"↑X" RETURN:      JMS         ECHO/ECHO AS UP ARROW AND CHAR
"IGNORE" RETURN:JMP           IGNORE
```

The user subroutine must obey the conventions of interrupt time routines: it must not call the Monitor, it must be short, and it must not call any routines which are also called by the background. It also must not attempt to do any I/O on the TTY.

1.6 Index Facilities

The LDP System tapes are $16ØØ_8$ ($896_{1Ø}$) blocks long, each block containing $4ØØ_8$ ($256_{1Ø}$) words. The first block is reserved for index and initial loading information.

All index entries have the following general form:

FILE NAME (6 CHARS)	File name is extended to 6 characters with blanks if necessary
FILE EXT	Period + (3 chars)
AUXILIARY	1 word
STARTING BLK	(12 bit unsigned integer)
LENGTH	(12 bit integer)

Empty index entries are filled with blanks in the name area. The auxiliary word is unused by the system, amd may therefore be used by user programs for whatever purpose is deemed appropriate.

There may be several index blocks on a device. A tape will ordinarily have four. In any case, the first three entries of the first index block (block Ø) are reserved and are used as follows:

Word	Contents
Ø-4	12$_8$ char tape ID
5	4ØØ$_8$ (block length)
6	4-(starting block of file area) = length of index
7	16ØØ$_8$ - (total blocks on device)
1Ø-25	Reserved for bootstrap program
26	First block of Job Control
27	Length of Job Control

2.0 ASSEMBLY INSTRUCTIONS

The AIPOS system source programs can be modified by the programmer to individualize his system. Assembly instructions for the system programs are included in this section. Note that a previous version of the program on the same LDP volume with the same name should be deleted just before the new version is created on that volume. All assembly instructions in this section are described in three steps:

1. Preparing the DIAL binary
2. Preparing the AIPOS binary
3. Manipulating the AIPOS index (this step is not always necessary).

2.1 Monitor

2.1.1 Preparing the DIAL Binary

The LDP Monitor consists of two CHAINed DIAL-MS[1] sources, MAØ1 and MBØ1. A LISTAPE 14 pseudo-op in MBØ1 facilitates cross-referencing of the files. Note that any modification to the Monitor requires reassembling the DIAL-MS Job Control file JLnn so that the JLnn file will have the correct Monitor symbols. Use the following sequence of DIAL-MS commands to assemble the Monitor:

```
→ZE ↲
→AS MAØ2,u ↲
→SB MBØ2,u ↲
```

where u is the appropriate DIAL unit and ↲ is carriage return.

2.1.2 Preparing the AIPOS Binary

Load the LDP system (refer to Appendix A of the AIPOS User's Manual, DEC-12-SQ1A-D) and then call the program INIT.

No index manipulation is required after INIT has been used for the Monitor.

[1] LAP6-DIAL-MS is referred to as DIAL-MS in this manual.

2.2 Job Control

2.2.1 Preparing the DIAL Binary

The Job Control Processor is a combination (via Add Binary) of three separate assemblies, the sources for which are JLØ2, CMØ2, XSAØ2 and XSBØ2 (Ø2 is the current version number for each source). Each source contains a LISTAPE 14 pseudo-op to allow the output to be CREF'ed. XSA contains a chain statement to XSB, which must be on the same unit.

The Job Control Processor, therefore, may be assembled by the following sequence of DIAL-MS commands:

```
→ZE↵
→AS JLØ2,u↵
→SB JLØ2,u↵

→ZE↵
→AS CMØ2,u↵
→SB CMØ2,u↵

→ZE↵
→AS XSAØ2,u↵
→SB XSAØ2,u↵

→AB CMØ2,u↵
→AB JLØ2,u↵
→SB JOBCTL,u↵
```

where u is the appropriate DIAL unit. Note that the Monitor must be reassembled before Job Control unless the SAVSYM area was saved from the last time the Monitor was assembled.

2.2.2 Preparing the AIPOS Binary

Load the LDP system and then call the program INIT.

After responding properly to INIT's messages, this version of the Job Control Processor can be run.

2.3 DORA

2.3.1 Preparing the DIAL Binary

DORA is comprised of fifteen sources: DC, DA, DA1, DA2, DA3, DB, OVRØ, OVR1, OVR2, OVR3, OVR4, OVR5, OVR6, ADA and ADB. The program is comprised of two main binaries (DA and DB) and a series of overlays. The LDP binary file DORA.BIN is created from the above DIAL sources. ADA and ADB are one-block sources which describe the assembly of the main sources. To assemble the DA binary, assemble the source ADA. The sources that define the binary DA are DC, DA, DA1, DA2, and DA3. These sources may be on any DIAL unit. The source ADA defines (via = statements) those units that contain the DA sources. For example: DA2U=11 means that the source DA2 is located on unit 11. To avoid chaining errors, ADA must be consistent with the actual location of the DA sources at assembly time. The binary DB is assembled by assembling the source ADB; the sources DC and DB define the binary DB. The overlay sources OVRØ-OVR6 and ADA and ADB have no unit restrictions.

Use the following sequence of DIAL-MS commands to assemble DORA:

```
     →ZE↲
     →AS ADA,u↲
     →SB DA,u↲

     →ZE↲
     →AS ADB,u↲
     →SB DB,u↲

     →ZE↲
     →AS OVRK,u↲        (K=Ø,.,...,6)
     →SB OVRK,u↲

     →ZE↲
     →AB DB,u↲
     →AB DA,u↲
     →SB DORA,u,PØ36ØØ↲
```

where u is the appropriate DIAL unit.

2.3.2 Preparing the AIPOS Binary

Load the LDP system. Delete a previous version of DORA if present. Then call the program BUILD by issuing a command in the form:

```
          dev:BUILD dev:DORA
```

Use the following information when responding to BUILD's messages:

primary binary	DORA
secondary binaries	OVRK for K=Ø, 1,...,6
scratch blocks	3
working areas	2

DORA can now be loaded and run. No AIPOS index manipulation is required.

2.4 File Handling Functions

2.4.1 Preparing the DIAL Binary

Three CHAINed sources comprise the file handling programs: FORCH, FORA, and FORB.

The symbol DA is defined as the FORA device. DB is defined as the FORB device. An example of a FORCH configuration is as follows:

```
          DA=11
          DB=Ø

          CHAIN "FORA" DA
```

The above configuration would be used if FORA were on unit 11 and FORB were on unit Ø.

Use the following sequence of DIAL-MS commands to assemble FORCH:

```
        →ZE ↵
        →AS FORCH,u ↵
        →SB FORCH,u,P1Ø2ØØ ↵
```

where u is the appropriate DIAL unit.

2.4.2 Preparing the AIPOS Binary

Load the LDP system and then call the program BUILD by issuing a command in the form

```
dev:BUILD  dev:CREATE
```

Use the following information when responding to BUILD's messages:

primary binary	FORCH
secondary binaries	Ø
scratch blocks	Ø
working areas	1

2.4.3 Manipulating the AIPOS Index

The other file handling functions are placed on the LDP volume using the Alias facility of DISPLAY INDEX. Call the DX function and then request the Alias option. From the existing file CREATE.BIN, create the five new file names:

```
INTERP.BIN
TRANS.BIN
PRINT.BIN
DISHDR.BIN
FIXHDR.BIN
```

2.5 BUILD

2.5.1 Preparing the DIAL Binary

The DIAL sources for BUILD (version 1) are assembled by the following sequence of DIAL-MS commands:

```
→|ZE↵
→AS BØ1,u↵
→SB BØ1,PØ4ØØØ ↵
```

where u is the appropriate unit.

2.5.2 Preparing the AIPOS Binary

If BUILD is to be created on the non-systems volume, proceed as follows. Load the LDP system and then call the program BUILD by issuing a command in the form:

```
dev:BUILD dev:BUILD
```

where the first part of the command is the function that will be used (systems device) and the second part is the new file that will be created. Use the following information when responding to BUILD's messages:

primary binary	BØ1
secondary binaries	Ø
scratch blocks	Ø
working areas	Ø

If the BUILD program being created is to replace the version currently on the systems volume, special precautions are required. The suggested procedure is as follows:

1. Call DISPLAY INDEX and use the delete facility to remove the file INIT.BIN. Then return to Job Control. This sequence appears on the Teletype as:

```
dev:DX↵
DINIT.BIN↵
↵
```

2. Call BUILD from Job Control and create the output file INIT:

```
dev:BUILD dev:INIT↵
```

3. Respond to the BUILD displays as follows:

```
BØ1,u↵
↵
↵
↵
```

where u is the DIAL unit containing the file INIT.

4. After returning to Job Control, request DISPLAY INDEX and rename the old BUILD file. Then Alias the INIT file to be the new BUILD file.

```
dev:DX↵
RBUILD.BIN↵
BUILD.BAK↵
AINIT.BIN↵
BUILD.BIN↵
```

When a new volume is initialized, it is often inconvenient to obtain the binary files to be stored on it from DIAL-MS files. In this case, it is preferable to use the MOVE facility to copy the binary files from one LDP volume to another. If, for instance, the programs BUILD, DORA, and MOVE (which are on disk Ø) are desired on the fresh initialized tape on unit 1, the following sequence of commands should be used:

```
MOVE LT1:BUILD.BIN=DKØ:BUILD.BIN↵
MOVE LT1:DORA.BIN=DKØ:DORA.BIN↵
MOVE LT1:MOVE.BIN=DKØ:MOVE.BIN↵
```

Note, too, that since INIT.BIN is merely an alias of BUILD.BIN, it is not necessary to MOVE it separately. It is sufficient to use the Alias facility of DISPLAY INDEX. Similarly, MOVE's aliases may be created on the new volume.

3.0 INTERNAL DESCRIPTION

This description is intended to be read with a listing of the Monitor.

3.1 Assembling the Monitor

Source of the Monitor consists of two DIAL-MS source files, respectively MAnn and MBnn, where nn is the current version number. MAnn contains a CHAIN statement for MBnn on the same unit. MBnn contains a LISTAPE 14 statement to facilitate cross-referencing. The Monitor is assembled and saved by the following DIAL commands.

```
    →ZE ↵
    →AS MAnn,u ↵
    →SB MBnn,u ↵
```

If the Monitor has been modified, Job Control module JLnn should be reassembled immediately after this, since correct system operation depends on JL having the correct Monitor symbols.

3.2 Calling Sequences

Standard calling sequence for all Monitor services is CLA (if AC is not clear,) then RIF; CIF Ø, (these two instructions are not necessary if the caller is in field zero), then a JMS indirect to the appropriate Monitor location, followed by the addresses of appropriate parameters. The first Monitor routine to be considered is SETINT, which is used by the operating program to enable an interrupt handler. A call to SETINT transfers control to location ISETR in the Monitor (approximately 6ØØ). ISETR calls MNTR, a Monitor subroutine which is used for saving status of the system at the time of the call. MNTR saves the current data field, the caller's field and the return address before returning to its caller. ISETR then searches the list of IOT skips beginning at INTSCN to find an IOT which matches the one specified by the caller. If the match is found, that IOT is replaced with a NOP (octal 7ØØØ). Then the address of the handler subroutine specified by the user is tested to see whether it is 7777. If it is, then this call is understood to be a request to delete an interrupt handler; otherwise, the handler address is stored into a list of addresses at INTADD and the IOT specified by the caller is moved into the Interrupt Scan Chain INTSCN. ISETR then returns via the subroutine MXIT which restores the state of the machine at the time of the call, pops the Monitor stack, and returns to the caller.

3.3 I/O Routines

The Read and Write routines, READR and WRITR, respectively, are called in a manner similar to that used for SETINT. In the location following the JMS is the address of a parameter list in the same field which has the standard format for all I/O requests. That format is: the first word contains unit indication in the internal form, the second word contains a CDF instruction pointing to the field containing the user's buffer, the third location is the address of the user's buffer within the specified field, the fourth location is the length of the buffer in words, the fifth location is the block number if the request refers to a Mass Storage Device. (It is ignored otherwise.) The sixth and seventh locations are used by the Monitor for queueing I/O requests. READR and WRITR then each call the subroutine MNTR. (The WRITR routine moves its return address to location .READR, and sets the Read/Write switch to -1. At this point, the two routines run in parallel at location IOCOM which stores the address of the user's parameter list in location TMP5, decodes the unit class (that is, bits 6-8 of the unit word) and jumps to the appropriate I/O subroutine as determined by ULIST, a list of I/O device handlers by unit class. Two subroutines, ENQ and DEQ, are used by almost all I/O handlers to control the queueing of I/O operations as required by the Monitor. ENQ is called with a pointer to the "queue pointer" by each device routine as it is necessary to queue each I/O request. ENQ examines the queue pointer (which is contained in the device routine) and takes a skip return if the I/O request is the first for this particular device, does not skip if there are other I/O operations in progress. In the latter case, the I/O request at this time is chained to preceding I/O requests such that it can be removed from the list as preceding requests are satisfied. Each queue pointer has the following format:

The first word is the CDF instruction pointing to the current I/O operation or a zero if there is no I/O operation in progress on the particular unit to which this queue pointer refers. The second word is the address of the current I/O list. The third word is a CDF instruction pointing to the buffer for this I/O request. The fourth word is the address of the buffer, updated appropriately as the transfer takes place. The fifth word is the number of words remaining to transfer in this operation, in 2's complement form, and the sixth word is the current block number for the transfer. The second, third, and fifth words, therefore, are direct copies of the corresponding information in the user's I/O request. The fourth word is the 2's complement of the user's request. The first two words are CDF instructions pointing to the field of the call and the second word is the parameter list address specified in the call.

ENQ turns interrupts off to prevent the possibility of an interrupt arising from an I/O request being satisfied while ENQ is in progress. The first word of the specified queue pointer is tested. If it is zero, then there is no activity on the particular device specified and appropriate parts of the parameter list are copied into the caller's queue pointer. If there is a CDF instruction in the first location of the queue pointer, then the queue is active and the new I/O request is chained to preceding I/O requests by storing CDF and the address of the new I/O request in the last two words of the preceding parameter list. ENQ then returns to the calling I/O handler. DEQ operates in approximately the reverse fashion. It is called, with interrupts off, at the time that a particular request has been satisfied. DEQ clears bit Ø of the unit word of the current parameter list, thereby indicating to the calling program that the I/O request has been satisfied. It then checks to find additional I/O requests in the chain for this particular device. If there is another request pending, it is moved to the queue pointer for the device and a skip return is taken from DEQ. If there is no further request for this device, then a short return is taken from DEQ indicating that all I/O operations on that device have been satisfied.

3.3.1 LINCtape Handler

We now turn our attention to the I/O handlers for specific devices, beginning with IOLTP, the LINCtape I/O routine. IOLTP is called by READR or WRITR whenever a request is in the unit class 2Ø, indicating that the request is for LINCtape. IOLTP calls ENQ, and if this is the first operation for LINCtape, a LINC instruction (6141) is moved to location LTLINC. The reason for this is explained in section 3.3.5, with the interrupt scan chain. The subroutine SET RW is called to set the Read/Write bit in the user's parameter list according to the status of RWSW, the Read/Write switch set by the Read/Write routine at the time of the call. Then if no other operation is in progress, the LINCtape subroutine LTPDO is called to start the requested operation.

When the operation has been started, the LINCtape I/O handler returns to the caller via routine MSXKP which restores the Monitor stack and takes a skip return at the site of the call. When the LINCtape processor causes an interrupt, control transfers to LTPINT (in LINCmode) which contains the code to handle LINCtape interrupts. At this time the LINCtape queue pointer LTPQP is tested to determine whether this interrupt indicates completion of the last transfer necessary to satisfy this request. If the transfer is complete then the subroutine DEQ is called to determine whether there are more requests on the list. If the transfer is not complete then LTPDO is called again to perform the next transfer. If DEQ indicates that there are no more requests pending for LINCtape, then routine LTICLR removes the LINCtape interrupt handler from the scan chain by storing a jump instruction at location LTLINC. It then clears the extended operations buffer to prevent multiple interrupts from LINCtape and returns through INTEX to the operating program at the point of the interrupt.

3.3.2 Teletype Handler

The Teletype I/O routine, IOTTY, is actually in two sections, one for input requests and one for output requests. Output requests are handled at TTOQ which calls the ENQ routine in a manner analogous to the way it was called by the LINCtape I/O handler, setting up the queue pointer for Teletype output, TTOQP. If there are no requests pending on the Teletype output queue pointer, the subroutine TTOPUT is called to print the first character from the buffer on the teleprinter. TTOPUT gets the character from the buffer, adjusts the pointer address, and calls subroutine TTECHO so that the output of the character is handled exactly as though it were the echo of an input character. Then, when an interrupt occurs from the teleprinter, the subroutine TTOINT (invoked by the interrupt scan chain) calls the echo routine to output the next character from the buffer. When the buffer has been exhausted TTOCLR stores a -1 in TTOFLG, indicating that no printing is currently active, clears the teleprinter flag with the instruction TCF and returns to the interrupt scan chain.

Subroutine TTECHO stores the character currently in the AC and checks to see whether the echo buffer is full. If it is, the character is ignored.

If the echo buffer is not full, the output flag TTOFLG is checked to determine whether the teleprinter is currently active. If not, the character is output. If it is, then this character is stored in the echo queue at TTEBSF. The pointers to the echo buffer are then adjusted and the subroutine TTECHO returns.

Teletype input requests are handled by the subroutine TTIQ which sets up the input queue pointer TTIQP by calling the subroutine ENQ. Teletype input interrupts are handled by the subroutine TTINT which gets the character, forces the parity bit to 1 and stores the character at TTICHR. Subroutine INTPSH is then called to push the interrupt status onto the interrupt stack so that interrupts can be turned on during processing of the Teletype interrupt. If the character is control (in the range 2ØØ-237), the user's control character handler (if any) is called. The character is compared to a list of characters in the special character table to determine whether it is one of those which requires special handling by the input routine. If it is not, the character is simply stored in the buffer and a control returns via INTPOP and then back through the interrupt scan chain. If the character is a special character (one of the following list):

377	RUBOUT
212 or 215	LINE FEED or CARRIAGE RETURN
233, 375, 376	ALTMODE
2Ø3	CONTROL C

one of the appropriate routines is invoked to handle that particular character. TTRUB, the routine which handles rubout, checks first to see whether input is active. If there is none active, control returns without echoing. If input is active and there are characters in the buffer, the last character in the buffer is cleared and a backslash is echoed on the console Teletype. If the character is 212 or 215 (line feed or carriage return), the routine TTCRLF verifies input activity, echoes carriage return/line feed, and stores the input characters in the user's buffer. The input request is considered satisfied and the DEQ is called to find the next request pending on the list. If the character is any of the three altmodes and input is active, dollar sign is echoed on the console Teletype and processing is the same as for carriage return or line feed. If the character is control C, the routine TTCTLC echoes an up-arrow and a C on the console Teletype, turns interrupts on, waits for the echo count to reach zero, and returns through EXIT. This subroutine does an I/O PRESET to clear all pending interrupts, and loads the job control processor from the system device.

3.3.3 Display Handler

The display I/O routine IODISP does not use subroutines ENQ or DEQ since it is not possible for a display request to be queued. In particular, display requests are satisfied immediately upon receipt of the request. Subroutine IODISP decodes the unit words in the user's parameter list. If bit 11 of that word is zero, then the request is display ASCII characters; if 1, the request is to display DIAL characters. Characters are obtained from the buffer, used to index the list of grids starting at location 3 in segment 1, and DSC instructions are used to display the appropriate grids. The characters in text format 43, 45 and 47 are considered to be special characters. 43 represents the new line character; 45 is the line feed without carriage return; and 47 is the tab, which takes the next character in the buffer to the next multiple of eight characters across the display screen.

3.3.4 Disk Handlers

Location IODISK is the start of the disk handlers. If RFØ8 and RK8 disks are both to be used with the system, then the routine IODISK first determines which type of disk is required for this particular request (indicates by bit 9 of the unit word). If bit 9 is set, the request is for RFØ8, otherwise it is for RK8. The RK8 routine enables its disk interrupt handler DSKINT by calling SETINT just as though it were a user program It then sets the read/write bit in the unit word of the user's parameter list and calls the subroutine DSKDO to start the I/O operation. DSKDO sets up the RK8 control registers and reads up to one full track of data from the disk. When the interupt from the disk occurs, subroutine DSKINT is called by the interrupt scan chain to process the interrupt. It determines whether this interrupt represents completion of the current request, whether the operation was completed without error and whether more operations are to be performed for subsequently queued requests. If an operation is to be performed, in each case the subroutine DSKDO is invoked to perform it. DEQ is called to get the next operation from the list when all operations have been performed, clear status is issued to clear the disk, and then the IOT in the interrupt scan chain is replaced with a NOP instruction and control returns to the interrupt scan chain. The RFØ8 routine is performed in a precisely analogous fashion for requests on RFØ8 disks.

3.3.5 Interrupt Status Storage

The interrupt time facilities for the Monitor are entered at location PINT or LINT depending on the mode of the machine at the time the interrupt occurred. Each of these routines saves the current AC and TINK registers along with an indication of the status of the machine, that is, whether it was in PDP-8 or LINCmode at the time of the interrupt. Then it saves the interrupted fields in location ISVFLD and enters INTSCN to begin searching for the source of the interrupt. INTSCN is a string of instructions, which at initialization time are JMPs and NOPs. Each NOP may be replaced by the subroutine ISETR with a VOT instruction which will cause a skip to a CIFCDF and a JMS indirect to the user's interrupt handler.

At the end of this chain are special interrupt handlers for the teleprinter, the keyboard, and LINCtape, which are special routines as far as the Monitor is concerned. The LINCtape routine must be handled in a particular way because the test has to be performed in LINCmode rather than in PDP-8 mode. To do this the instruction at LTLINC may be either an indirect JMP to return to INTEX, the interrupt exit routine, or may be a LINC instruction (6141). When tape operations are in progress, if this is a LINC instruction, control flows in to a "skip on tape not done" (STD I) instruction, which, if tape is done, permits a JMP to LTPINT, the LINCtape interrupt handler to take place. Otherwise, control flows into a PDP instruction then jump indirect to INTEX. At INTEX, the interrupt exit routine, the fields as saved at ISVFLD are restored through LDF and LIF or CIF instructions to return the fields to their status at the time of the interrupt. Then the AC and the LINC are restored; and the machine state, either PDP mode or LINC mode, is restored; and control returns, after turning interrupts on, to the program at its point of interruption.

Two more routines must be examined at this point, IPUSHR and IPOPR, the two routines which push and pop the interrupt status on the interrupt stack. These two routines simply interrogate the values of the saved AC, the saved LINK and MODE, and the return address words and push them according to the values of Auto-index register 16, then at pop time, pop them off the list.

3.4 Bootstrap Routine

The initial loading bootstrap routine is assembled at location 6ØØØ in the Monitor. This routine is the first block loaded when the Monitor is initially loaded from tape or disk. If the load is from disk, this block may be loaded at or below location 4ØØ1 and all addresses are one higher than the assembled address because of the fact that the word count and current address used to load the system from disk are actually in-correct. (I/O PRESET clears the word count and current address registers in the disk control. Word Ø of block Ø, therefore, is loaded into location 1, etc.) If the loading program is loaded from disk, the entry point is that assembled at location 6ØØØ, but the load routine calls subroutine IMOVE to move the entire load routine to location 6ØØØ. Otherwise, tape and disk load are identical following location INFOR, which collects information from the index which was also loaded at startup time. Then the Monitor is read in using special subroutines which are contained in the loader program. Upon completion of this operation, page 37 is cleared to prevent files from being saved improperly, and the loading program jumps to location EXIT which causes further processing just as though a normal program has completed operation.

3.5 Implementing Additional Devices

Using a listing of the Monitor, examine the coding of one of the present device routines. If the device uses data break for transfers, look particularly at the disk routines. If transfers take place through the AC, the TTY routines will be more useful.

3.5.1 Accessing the Device

If the device is in one of the system-defined classes, turn on the conditional assembly control for that class, to allow the READ and WRITE routines to access the device routine. If there are other devices in the same class, add code at the device routine to decode the low-order bits of the unit code for appropriate handling of the various devices in the class.

The device routine (assuming the device generates interrupts) must have two sections: one to queue the request and start it, if there are no other requests on the queue; the other to handle the interrupts, continue the operation if incomplete, retry errors if appropriate, and dequeue completed requests.

3.5.2 Queueing

Queueing is performed by the NQUEUE routine,which sets the data field to the user's calling field, turns interrupts off, and chains the current request to the other requests on this device the parameter list for this request is copied into the queue pointer for this device, and the second return from NQUEUE is taken.

This queue pointer for any device contains information about the current operation taking place on a device. The first word is zero if the device is idle. If the device is busy, the first word contains a CDF instruction for the field containing the parameter list for the current operation, and the second word contains the address within the field of that parameter list. The next two words specify the field and address of the buffer for this operation. The next word is the two's complement of the length specified in the caller's parameter list, and the final word is the starting block of the operation, if tape or disk. Non-block oriented device routines should ignore this last word.

3.5.3 Dequeueing

Dequeueing I/O requests is performed by the routine DQUEUE, which clears the busy bit in the user's parameter list, clears the chain pointer in his list, and copies the parameter list into the queue pointer for the device. If there is no pending I/O request for the device, the alternate return from DQUEUE is taken, so that the device routine can clear the device flags and enter the idle state, rather than attempting to start a new operation.

WARNING: Dequeueing must not be performed until an operation is fully complete, i.e., the device routine must not dequeue at the time it starts the last phase (last block or last character) of a transfer, but wait until the "done" flag comes up at the device before dequeueing.

3.5.4 Interrupts

If a device routine services a low-priority device, or takes a significant amount of time to service it, the routine should make use of the INTPSH and INTPOP facilities to allow rapid service of high-priority interrupts. If this is done, remember that the device flag for the device being serviced must be cleared before turning interrupts on, and that DQUEUE must be called with interrupts off.

New device routines should use the SETINT facility to set up the call to their interrupt handlers. When the device is idle, the interrupt handler may disable itself by storing a NOP (7ØØØ) at its return address minus 4. (The disk routines use this technique to allow the flag-test slot to be available for other SETINTs during the time the device is idle.)

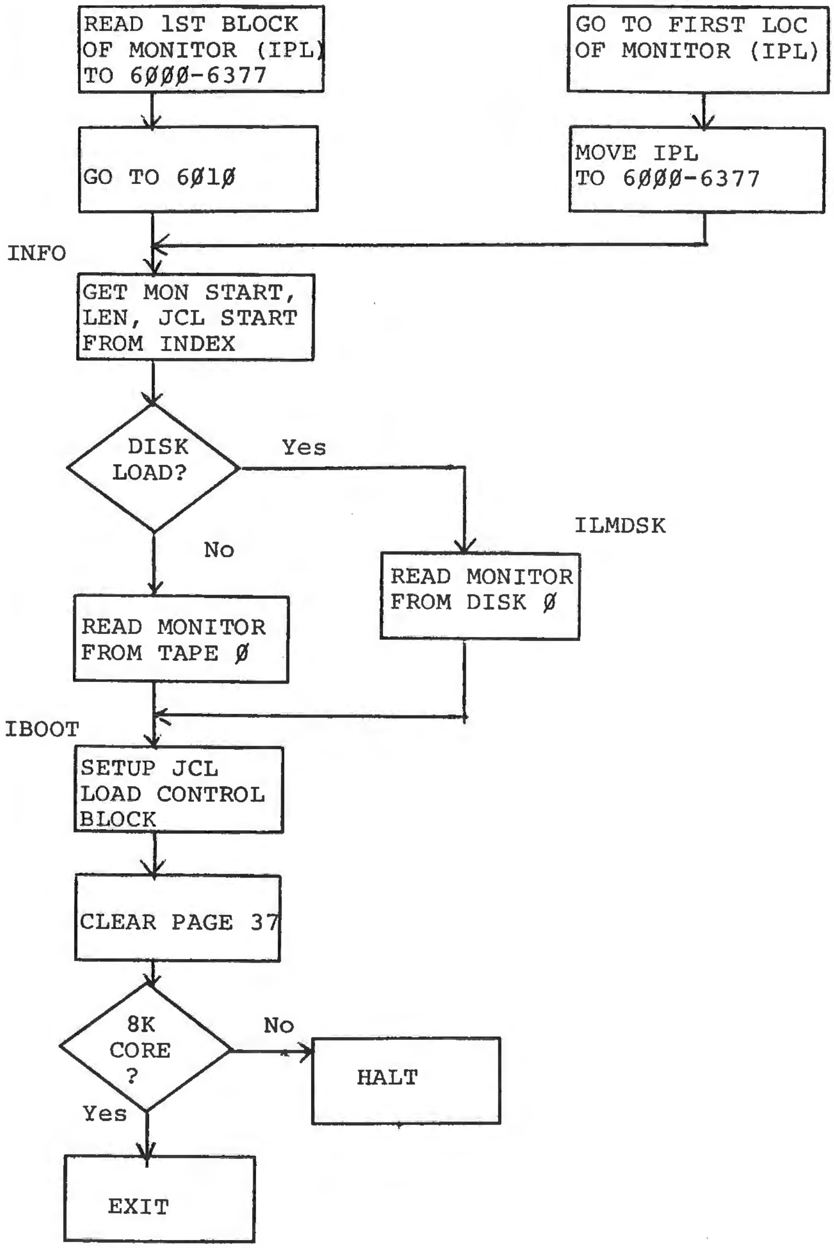
4.0 MONITOR CORE MAP

FIELD Ø:	Ø-1	8-MODE INTERRUPT
	2-7	MONITOR TEMPORARIES
	1Ø-15	USER AUTO-INDEX REGISTERS
	16-17	MONITOR AUTO-INDEX
	2Ø-37	MONITOR CALL ADDRESSES
	4Ø-43	LINC-MODE INTERRUPT
	44-137	USER PAGE Ø
	14Ø-177	MONITOR PAGE Ø
	2ØØ-377	INTERRUPT PROCESSING
	4ØØ-577	INTERRUPT SCAN CHAIN AND VARIOUS TABLES
	6ØØ-777	ISETR, UTILITY ROUTINES
	11ØØ-1177	READR, WRITR, ENQ, DEQ
	12ØØ-1377	LINCTAPE ROUTINE
	14ØØ-1577	TTY OUTPUT, ECHO
	16ØØ-1777	TTY INPUT
	2ØØØ-22ØØ	DISPLAY CHAR TABLE
	22Ø1-2377	CHAR DISPLAY ROUTINE
	24ØØ-2577	DISK I/O ROUTINE
	3ØØØ-76ØØ	USER PROGRAM AREA
	76ØØ-7777	JCL/USER COMMUNICATION
FIELD 1:	Ø-7777	USER PROGRAM AREA

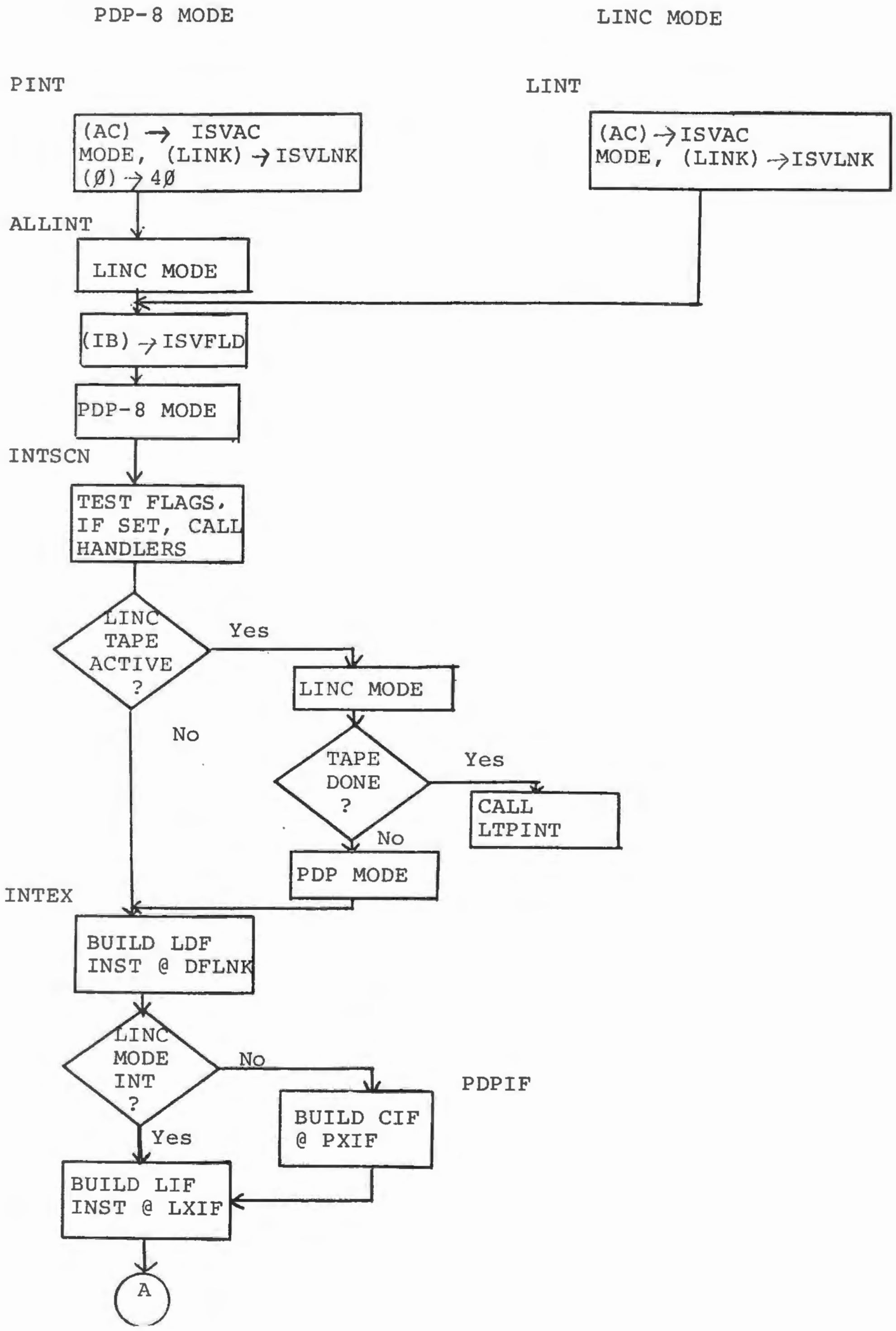
5.0 FLOW CHARTS

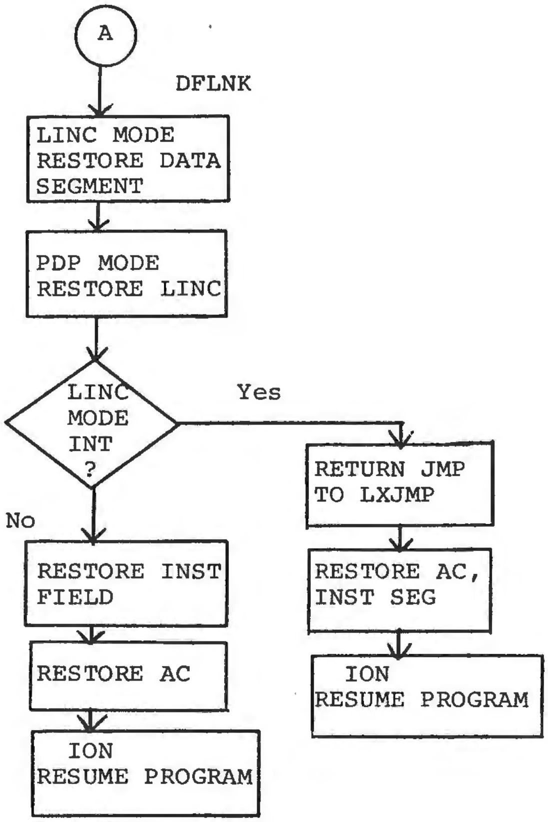
5.1 Initial Program Load

Tape:	LSW=7ØØ	Disk:	2Ø = 6733
	RSW=Ø		21 = 5Ø21
	LINC mode		PDP-8 mode
	I/O PRESET		I/O PRESET
	DO		START 2Ø
	START 2Ø		

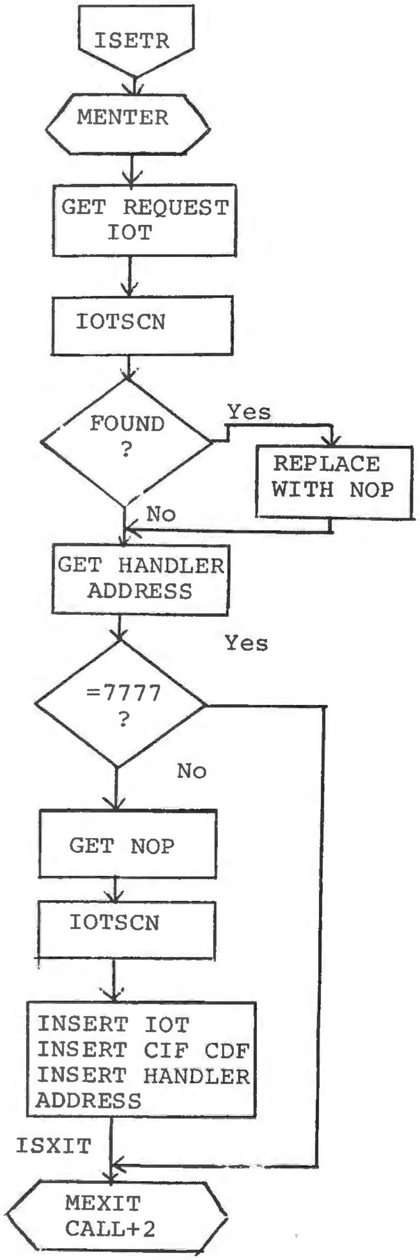

READ 1ST BLOCK OF MONITOR (IPL) TO 6ØØØ-6377
GO TO 6Ø1Ø
GO TO FIRST LOC OF MONITOR (IPL)
MOVE IPL TO 6ØØØ-6377
INFO
GET MON START, LEN, JCL START FROM INDEX
DISK LOAD?
Yes
No
ILMDSK
READ MONITOR FROM DISK Ø
READ MONITOR FROM TAPE Ø
IBOOT
SETUP JCL LOAD CONTROL BLOCK
CLEAR PAGE 37
8K CORE ?
No
HALT
Yes
EXIT

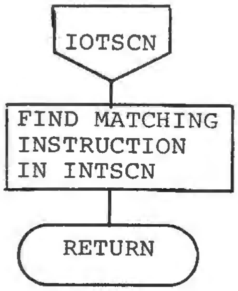
5.2 Interrupt Processing

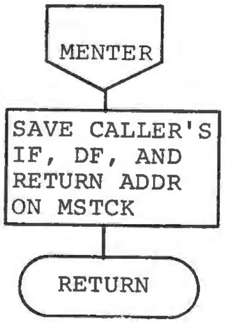


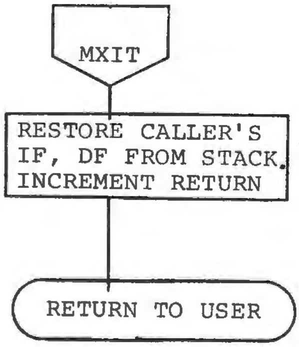

A
DFLNK
LINC MODE
RESTORE DATA
SEGMENT
PDP MODE
RESTORE LINC
LINC
MODE
INT
?
Yes
No
RETURN JMP
TO LXJMP
RESTORE AC,
INST SEG
ION
RESUME PROGRAM
RESTORE INST
FIELD
RESTORE AC
ION
RESUME PROGRAM

5.3 SETINT

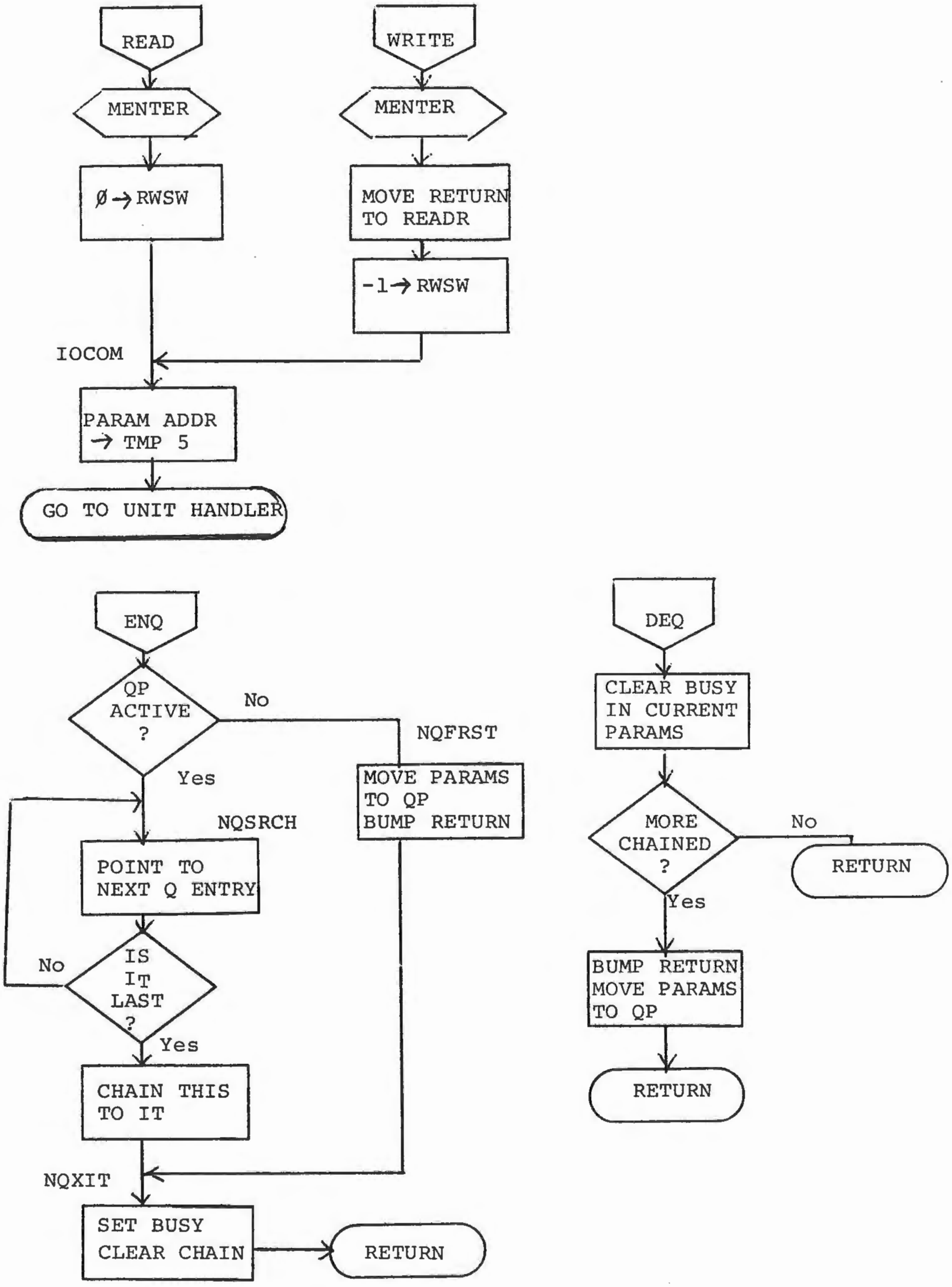

ISETR
MENTER
GET REQUEST IOT
IOTSCN
FOUND ?
Yes
REPLACE WITH NOP
No
GET HANDLER ADDRESS
Yes
=7777 ?
No
GET NOP
IOTSCN
INSERT IOT
INSERT CIF CDF
INSERT HANDLER ADDRESS
ISXIT
MEXIT CALL+2


IOTSCN
FIND MATCHING INSTRUCTION IN INTSCN
RETURN

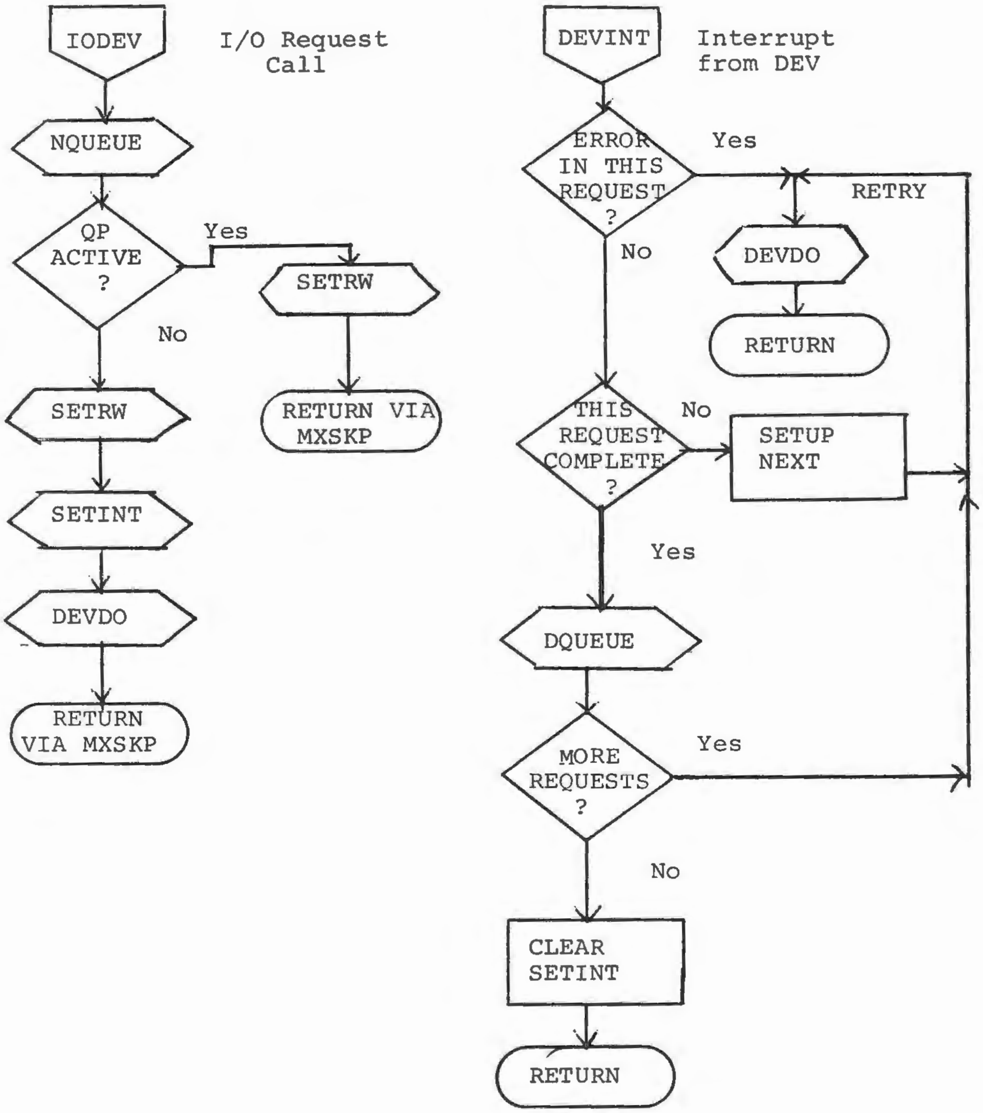

MENTER
SAVE CALLER'S IF, DF, AND RETURN ADDR ON MSTCK
RETURN


MXIT
RESTORE CALLER'S IF, DF FROM STACK.
INCREMENT RETURN
RETURN TO USER

5.4 General I/O Requests



5.5 General Form of All Device Handlers



```
*2Ø
/       LDP SYSTEM RESIDENT MONITOR
/
/COPYRIGHT 1971; DIGITAL EQUIPMENT CORP,
/                MAYNARD, MASS, Ø1754
/
/       VERSION 1: FEBRUARY 23, 1971
/               JUD LEONARD
/
        PMODE
        SAVSYM  2       /SAVE SYMBOL TABLE FOR JCL
/
/       JOB CONTROL MODULE JLNN MUST BE
/       REASSEMBLED WHEN MONITOR IS MODIFIED
/
INTMAX= 1Ø              /MAXIMUM NUMBER OF INTERRUPT
                        /HANDLERS ENABLED
                        /MUST NOT EXCEED 26
INTSTK= 4               /MAXIMUM INTRPT PUSHDOWN DEPTH
/
MONSTK= 4               /MAX DEPTH OF MONITOR STACK
/
/       ADJUSTMENT OF THE ABOVE VARIABLES MAY CAUSE
/       AN INCREASE IN THE SIZE OF THE MONITOR,
/       EXAMINE THE CODING FOLLOWING
/       INTSCN TO DETERMINE OPTIMUM VALUES,
/
/       THE FOLLOWING VARIABLES ARE INTENDED FOR
/       CONDITIONAL ASSEMBLY CONTROL OF DEVICE
/       ROUTINES, MANY OF THE ASSOCIATED ROUTINES
/       DO NOT EXIST, OR ARE NOT DEBUGGED,
/       THEREFORE USE GREAT CAUTION IN CHANGING
/       THESE VARIABLES,
/
/
RKØ8=   1               /NUMBER OF RK8 DISK DRIVES
RFØ8=   Ø               /NUMBER OF RSØ8 DISK PLATTERS
PC12=   Ø               /1 FOR PAPER TAPE
LPØ8=   Ø               /1 FOR LPØ8 PRINTER
CR12=   Ø               /1 FOR CARD SUPPORT
TC58=   Ø               /1 FOR MAG TAPE SUPPORT
MSCDEV= PC12!LPØ8!CR12!TC58
KP12=   Ø               /1 FOR POWER FAIL SUPPORT
DISPLY= 1               /1 FOR CHAR DISPLAY ROUTINES
GRIDS=  Ø               /1 FOR 4*6 ASCII DISPLAY GRIDS
/
TTBLEN= 4               /TTY ECHO BUFFER LENGTH
                        /MUST BE POWER OF TWO
/
DIAL=   Ø               /1 FOR DIAL-MS TESTING
/                       /VIA ADD BINARY
CREF=   14              /LISTAPE UNIT IF NOT ZERO
/
/
        EJECT
```

```
                /
                        ASMIFZ  DIAL
                        FIELD   1         /PUT IPL BEFORE MONITOR
                /                         IN BINARY FILE
                        *Ø
                /
                /
                /       PDP=8 MODE INTERRUPT
                /
ØØØØ  ØØØØ              Ø
ØØØ1  5544              JMP I    P8INT   /GO TO INTERRUPT HANDLER
                /
                /       NON=INTERRUPT TEMPORARIES
                /
ØØØ2  ØØØØ      TMP1,   Ø
ØØØ3  ØØØØ      TMP2,   Ø
ØØØ4  ØØØØ      TMP3,   Ø
ØØØ5  ØØØØ      TMP4,   Ø
ØØØ6  ØØØØ      TMP5,   Ø
ØØØ7  ØØØØ      TMP6,   Ø
                /
                /       AUTO-INDEX REGISTERS 1Ø-15
                /       RESERVED FOR USER PROGRAMS
                /
                        *16
ØØ16  Ø467      ISTP,   ISTACK  /START OF INTERRUPT PUSH-DOWN LIST
ØØ17  Ø51Ø      MSTP,   MSTACK  /START OF MONITOR CALL PUSH-DOWN LIST
                /
                /
                /       MONITOR CALL ADDRESSES
                /
                /
ØØ2Ø  Ø6ØØ      SETINT, ISETR             /SETUP INTERRUPT HANDLER
ØØ21  1ØØØ      READ,   READR             /READ ROUTINE
ØØ22  1ØØ3      WRITE,  WRITR             /WRITE ROUTINE
ØØ23  Ø715      WAIT,   WAITR             /SINGLE WAIT
ØØ24  ØØ31      MWAIT,  MHALT             /MULTIPLE WAIT
ØØ25  1335      EXIT,   JCBOOT            /RETURN TO JOB CONTROL
ØØ26  ØØ33      ABORT,  MNOP              /DE-QUEUE I/O
ØØ27  Ø212      INTPSH, IPUSHR            /PUSH INTERRUPT ONTO STACK
ØØ3Ø  Ø236      INTPOP, IPOPR             /POP SAME
                /
ØØ31  ØØØØ      MHALT,  Ø                 /UNIMPLEMENTED CALLS
ØØ32  5552              JMP I   MERROR
ØØ33  ØØØØ      MNOP,   Ø                 /NOP CALLS
ØØ34  4547              JMS I   MENTER
ØØ35  555Ø              JMP I   MXSKP
                /
                /       HELPFUL VALUES
                /
ØØ36  ØØØ2              Ø2                /CURRENT VERSION
ØØ37  26ØØ              MEND+1            /FIRST USER LOC
                /
                /       LINC-MODE INTERRUPT
                /
                        LMODE
                        *4Ø
ØØ4Ø  ØØØØ              Ø
ØØ41  Ø5ØØ              IOB
ØØ42  6ØØ2              LIOF              /PREVENT   TIPLE INTERRUPTS
ØØ43  6366              JMP     LINT      /HANDLE    ERRUPT
                /
                        EJECT
```

```
            /
            /       LOCATIONS 44 THRU 137 AVAILABLE
            /       TO USER PROGRAMS
            /
                    LMODE
                    *140
0140  0000          0
0141  0000          HLT             /TRAP OR 6141
0142  0002          PDP
                    PMODE
0143  4425          JMS I   EXIT    /ERR RTRN TO JCL
            /
            /       LOCATIONS 144 THRU 177 AVAILABLE TO MONITOR ONLY
            /
0144  0360  P8INT,  PINT            /8-MODE INTERRUPT HANDLER
0145  1030  NQUEUE, ENQ             /QUEUE UPDATE ROUTINE
0146  1103  DQUEUE, DEQ             /DOWN DATE
0147  0725  MENTER, MNTR            /GENERAL ENTRY FOR MONITOR CALLS
0150  0751  MXSKP,  MX1             /EXIT WITH SKIP
0151  0746  SETUDF, MUDF            /CDF TO USER
0152  0677  MERROR, MERR            /MONITOR ERROR
            /
0153  0070  P70,    70
0154  0077  P77,    77
0155  0005  P5,     5
0156  7774  M4,     -4
0157  6201  KCDF,   CDF
0160  7000  KNOP,   NOP
0161  0260  CZERO,  260
            /
            RWSW=   TMP6            /0 IF READ, -1 IF WRITE
            /
            /       TEMPORARIES FOR USE BY INTERRUPT-TIME ROUTINES
            /       MUST BE USED ONLY WITH INTERRUPTS OFF
            /
0162  0000  QT1,    0
0163  0000  QT2,    0
0164  0000  QT3,    0
0165  0000  QT4,    0
            /
                    ASMIFN  KP12
            PWRUPP, PWRUP           /POWER RESTORED
            /
                    *177
0177  1637  TTPCTL, TTINRM-1        /DEFAULT CNTRL CHAR HANDLER
            /
            LIOF=   IOF
            LION=   ION
            LRIB=   RIB
            /
            ONE=    CLA IAC
            TWO=    CLA CLL IAC RAL
            THREE=  CLA STL IAC RAL
            FOUR=   CLA CLL IAC RTL
            SIX=    CLA STL IAC RTL
            MONE=   CLA CMA
            MTWO=   CLA CLL CMA RAL
            MTHREE= CLA CLL CMA RTL
            /
                    EJECT
```

```
                   /
                   /
                           PAGE     1
                   /
                   /       INTERRUPT SERVICE ROUTINES
                   /
Ø2ØØ   ØØØØ        NTR,    Ø
Ø2Ø1   Ø153                AND     P7Ø
Ø2Ø2   1157                TAD     KCDF
Ø2Ø3   7ØØ1                IAC             /MAKE CIF
Ø2Ø4   3231                DCA     XITIF
Ø2Ø5   6214                RDF
Ø2Ø6   1157                TAD     KCDF
Ø2Ø7   323Ø                DCA     XITDF
Ø21Ø   62Ø1                CDF     Ø
Ø211   56ØØ                JMP I   NTR
                   /
                   /
                   /       PUSH INTERRUPT STATUS
                   /
Ø212   ØØØØ        IPUSHR, Ø               /INTERRUPT PUSH
Ø213   42ØØ                JMS     NTR
Ø214   1Ø16                TAD     ISTP    /CHECK FOR TOO
Ø215   1265                TAD     ISTLIM  /MANY PUSHES
Ø216   77ØØ                SMA CLA
Ø217   5234                JMP     IPERR1  /OOPS
Ø22Ø   1333                TAD     ISVAC   /SAVED AC
Ø221   3416                DCA I   ISTP
Ø222   1345                TAD     ISVLNK  /LINK AND MODE
Ø223   3416                DCA I   ISTP
Ø224   1Ø4Ø                TAD     4Ø      /RETURN ADDR
Ø225   3416                DCA I   ISTP
Ø226   1346                TAD     ISVFLD  /FIELDS
Ø227   3416                DCA I   ISTP
                   /
                   /
Ø23Ø   62Ø1        XITDF,  CDF
Ø231   62Ø2        XITIF,  CIF
Ø232   5612                JMP I   IPUSHR  /RETURN
                   /
Ø233   72Ø1        IPERR2, ONE
Ø234   7ØØ1        IPERR1, IAC
Ø235   5552                JMP I   MERROR
                   /
                           EJECT
```

```
                    /
                    /
                    /       POP INTERRUPT STATUS
                    /
      Ø236  ØØØØ    IPOPR,  Ø
      Ø237  42ØØ            JMS     NTR
      Ø24Ø  1156            TAD     M4
      Ø241  1Ø16            TAD     ISTP
      Ø242  3Ø16            DCA     ISTP
      Ø243  1Ø16            TAD     ISTP    /CHECK FOR TOO
      Ø244  1266            TAD     MISTAK  /MANY POPS
      Ø245  771Ø            SPA CLA
      Ø246  5233            JMP     IPERR2  /OOPS
      Ø247  1416            TAD I   ISTP
      Ø25Ø  3333            DCA     ISVAC   /POP SAVED AC
      Ø251  1416            TAD I   ISTP
      Ø252  3345            DCA     ISVLNK  /LINK
      Ø253  1416            TAD I   ISTP
      Ø254  3Ø4Ø            DCA     4Ø      /RETURN ADDR
      Ø255  1416            TAD I   ISTP
      Ø256  3346            DCA     ISVFLD  /FIELDS
      Ø257  1156            TAD     M4
      Ø26Ø  1Ø16            TAD     ISTP
      Ø261  3Ø16            DCA     ISTP
      Ø262  1236            TAD     IPOPR
      Ø263  3212            DCA     IPUSHR
      Ø264  523Ø            JMP     XITDF
                    /
      Ø265  7271    ISTLIM, -ISTACK-INTST4
      Ø266  7311    MISTAK, -ISTACK
                    /
                            EJECT
```

```
            /
            /       GENERAL INTERRUPT EXIT
            /
Ø267  72ØØ  INTEX,  CLA
Ø27Ø  1346          TAD     ISVFLD  /GET INTERRUPTED FIELDS
Ø271  Ø347          AND     P37     /DATA SEGMENT IN 7-11
Ø272  1351          TAD     KLDF    /MAKE LDF INST
Ø273  3317          DCA     DFLNK+1 /FOR LATER EXECUTION
Ø274  73Ø7          FOUR
Ø275  Ø345          AND     ISVLNK  /MODE IS BIT 9
Ø276  764Ø          SZA CLA         /SKIP IF LINC MODE
Ø277  531Ø          JMP     PDPIF   /ELSE 8-MODE
            /
            /       BUILD LIF INSTRUCTION FOR LINC MODE RETURN
            /
Ø3ØØ  1346          TAD     ISVFLD  /FIELDS
Ø3Ø1  7Ø12          RTR
Ø3Ø2  7Ø12          RTR
Ø3Ø3  7Ø1Ø          RAR             /5-BIT IF TO 7-11
Ø3Ø4  Ø347          AND     P37     /INST SEG ONLY
Ø3Ø5  135Ø          TAD     KLIF    /BUILD LIF INST
Ø3Ø6  3334          DCA     LXIF    /LINC EXIT INSTRUCTION FIELD
Ø3Ø7  5316          JMP     DFLNK   /GO SET LINK AND DATA FIELD
            /
            /       BUILD CIF INSTRUCTION FOR PDP MODE RETURN
            /
Ø31Ø  1346  PDPIF,  TAD     ISVFLD  /GET SAVED FIELDS
Ø311  7Ø12          RTR
Ø312  7Ø12          RTR             /HIGH ORDER IF TO BITS 6-8
Ø313  Ø153          AND     P7Ø     /DROP OTHERS
Ø314  1352          TAD     KCIF    /BUILD CIF INST
Ø315  3341          DCA     PXIF    /PDP EXIT INST FIELD
            /
            /       RESTORE DATA FIELD AND LINK
            /
Ø316  6141  DFLNK,  LINC            /DO THE LDF
Ø317  ØØØØ          Ø               /LDF GOES HERE
Ø32Ø  ØØØ2          ØØØ2            /RETURN TO PMODE
Ø321  1345          TAD     ISVLNK  /GET LINK AND MODE
Ø322  7Ø12          RTR             /SET LINK
Ø323  764Ø          SZA CLA         /LINC MODE?
Ø324  5341          JMP     PXIF    /NO-EXIT IN PDP
            /
                    EJECT
                    /
                    /       EXIT FROM LINC MODE INTERRUPT
                    /
Ø325   1Ø4Ø                 TAD     4Ø       /RETURN ADDR
Ø326   6141                 LINC
                            LMODE
Ø327   162Ø                 BSE I            /FORCE JMP INST
Ø33Ø   6ØØØ                 JMP
Ø331   434Ø                 STC     LXJMP    /FOR RETURN
Ø332   1Ø2Ø                 LDA I            /RESTORE AC
Ø333   ØØØØ         ISVAC,  Ø
Ø334   Ø6ØØ         LXIF,   LIF              /RESTORE INST SEG
Ø335   ØØØ6                 DJR
Ø336   Ø5ØØ                 IOB              /RESTORE INTERRUPTS
Ø337   6ØØ1                 LION
Ø34Ø   6ØØØ         LXJMP,  JMP              /RESUME EXECUTION
                    /
                    /       EXIT FROM PMODE INTERRUPT
                    /
                            PMODE
                    /
Ø341   62Ø2         PXIF,   CIF              /RESTORE INST FLD
Ø342   1333                 TAD     ISVAC    /RESTORE AC
Ø343   6ØØ1                 ION
Ø344   544Ø                 JMP I   4Ø       /RESUME USER PGM
                    /
                    /
Ø345   ØØØØ         ISVLNK, Ø
Ø346   ØØØØ         ISVFLD, Ø
                    /
Ø347   ØØ37         P37,    37
Ø35Ø   Ø6ØØ         KLIF,   6ØØ
Ø351   Ø64Ø         KLDF,   64Ø
Ø352   62Ø2         KCIF,   CIF
                    /
                            EJECT
```

```
            /
                    *36Ø    /FOLLOWING CODE AT EXACT END OF PAGE
            /
            /       ENTER HERE ON PDP-8 MODE INTERRUPT
            /
            /
Ø36Ø  3333  PINT,   DCA     ISVAC   /SAVE AC
Ø361  7007          IAC RTL         /SET MODE, SAVE LINK
Ø362  3345          DCA     ISVLNK
Ø363  1ØØØ          TAD     Ø       /RETURN ADDRESS
Ø364  3Ø4Ø          DCA     4Ø
Ø365  5372          JMP     ALLINT  /GO TO COMMON HANDLER
            /
            /
            /       ENTER HERE ON LINC MODE INTERRUPT
            /
                    LMODE
            /
Ø366  4333  LINT,   STC     ISVAC   /SAVE AC
Ø367  Ø262          ROL I   2       /SAVE LINC
Ø37Ø  4345          STC     ISVLNK
Ø371  Ø456          SKP
Ø372  6141  ALLINT, 6141            /LINC MODE FOR GETTING FIELDS
Ø373  Ø5ØØ          IOB
Ø374  6234          LRIB            /SAVE FIELDS
Ø375  Ø242          ROL     2       /IF IN 2-6, DF IN 7-11
Ø376  4346          STC     ISVFLD
Ø377  ØØØ2          PDP
                    PMODE
            /
                    ASMIFN  ,&177   /CHECK FOR PAGE BOUNDARY
            ASSEMBLY ERROR  /ADJUST ORG BEFORE "PINT"
                            /TO PUT THIS AT PAGE BOUNDARY
            /
                    EJECT
```

```
/
/       SCAN FLAGS TO FIND SOURCE OF INTERRUPT
/
/       THERE MAY BE A MAXIMUM OF 30 OCTAL ENTRIES IN
/       THE SCAN, AND THEY MUST ALL BE IN THE SAME PAGE
/
/
INTSCN= .
/
/
        INTMX4=INTMAX+INTMAX+INTMAX+INTMAX
        ASMSKP  156-INTMX4-INTMAX
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-27
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-26
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-25
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-24
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-23
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-22
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-21
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-20
/
        NOP
        JMP     .+3
        CIF CDF
        JMS I   INTADD-17
/
        NOP
        JMP     .+3
```

```
Ø661                              NOP
Ø662                              JMP       .+3
Ø663                              CIF CDF
Ø664                              JMS I     INTADD-15
Ø665                       /
Ø666                              NOP
Ø667                              JMP       .+3
Ø67Ø                              CIF CDF
Ø671                              JMS I     INTADD-14
Ø672                       /
Ø673                              NOP
Ø674                              JMP       .+3
Ø675                              CIF CDF
Ø676                              JMS I     INTADD-13
Ø677                       /
Ø7ØØ                              NOP
Ø7Ø1                              JMP       .+3
Ø7Ø2                              CIF CDF
Ø7Ø3                              JMS I     INTADD-12
Ø7Ø4                       /
Ø7Ø5       Ø4ØØ   7ØØØ            NOP
Ø7Ø6       Ø4Ø1   52Ø4            JMP       .+3
Ø7Ø7       Ø4Ø2   62Ø3            CIF CDF
Ø71Ø       Ø4Ø3   4656            JMS I     INTADD-11
Ø711                       /
Ø712       Ø4Ø4   7ØØØ            NOP
Ø713       Ø4Ø5   521Ø            JMP       .+3
Ø714       Ø4Ø6   62Ø3            CIF CDF
Ø715       Ø4Ø7   4657            JMS I     INTADD-1Ø
Ø716                       /
Ø717       Ø41Ø   7ØØØ            NOP
Ø72Ø       Ø411   5214            JMP       .+3
Ø721       Ø412   62Ø3            CIF CDF
Ø722       Ø413   466Ø            JMS I     INTADD-7
Ø723                       /
Ø724       Ø414   7ØØØ            NOP
Ø725       Ø415   522Ø            JMP       .+3
Ø726       Ø416   62Ø3            CIF CDF
Ø727       Ø417   4661            JMS I     INTADD-6
Ø73Ø                       /
Ø731       Ø42Ø   7ØØØ            NOP
Ø732       Ø421   5224            JMP       .+3
Ø733       Ø422   62Ø3            CIF CDF
Ø734       Ø423   4662            JMS I     INTADD-5
Ø735                       /
Ø736       Ø424   7ØØØ            NOP
Ø737       Ø425   523Ø            JMP       .+3
Ø74Ø       Ø426   62Ø3            CIF CDF
Ø741       Ø427   4663            JMS I     INTADD-4
Ø742                       /
Ø743       Ø43Ø   7ØØØ            NOP
Ø744       Ø431   5234            JMP       .+3
Ø745       Ø432   62Ø3            CIF CDF
Ø746       Ø433   4664            JMS I     INTADD-3
Ø747                       /
Ø75Ø       Ø434   7ØØØ            NOP
Ø751       Ø435   524Ø            JMP       .+3
Ø752       Ø436   62Ø3            CIF CDF
Ø753       Ø437   4665            JMS I     INTADD-2
                           /
Ø755                              EJECT
```

```
                             /
Ø757       Ø44Ø  6Ø41                TSF                 /PRINTER ALWAYS ENABLED
Ø76Ø       Ø441  5244                JMP       .+3
Ø761       Ø442  62Ø3                CIF CDF
Ø762       Ø443  4666                JMS I     INTADD-1
Ø763                         /
Ø764       Ø444  6Ø31                KSF                 /KBD ALWAYS ACTIVE
Ø765       Ø445  525Ø                JMP       .+3
Ø766       Ø446  62Ø3                CIF CDF Ø
Ø767       Ø447  4667                JMS I     INTADD
Ø77Ø                         /
Ø771                         /       LINC TAPE INTERRUPT TEST MUST BE LAST IN LIST
Ø772                         /
Ø773       Ø45Ø  5655        LTLINC, JMP I     LTNONE+1/LINC INSTR
Ø774                                 LMODE               /WHEN TAPE ACTIVE
Ø775       Ø451  Ø436                STD I               /IS TAPE DONE?
Ø776       Ø452  7275                JMP       LTPINT    /YES
Ø777       Ø453  ØØØ2                PDP                 /NO
1ØØØ                                 PMODE
1ØØ1       Ø454  5655        LTNONE, JMP I     .+1       /RESTORE STATUS
1ØØ2       Ø455  Ø267                INTEX               /AND RESUME EXECUTION
1ØØ3                         /
1ØØ4                         /
1ØØ5                         /       ADDRESSES OF INTERRUPT HANDLERS
1ØØ6                         /
1ØØ7                                 ASMSKP    26-INTMAX
1Ø1Ø                                 Ø
1Ø11                                 Ø
1Ø12                                 Ø
1Ø13                                 Ø
1Ø14                                 Ø
1Ø15                                 Ø
1Ø16                                 Ø
1Ø17                                 Ø
1Ø2Ø                                 Ø
1Ø21                                 Ø
1Ø22                                 Ø
1Ø23                                 Ø
1Ø24                                 Ø
1Ø25                                 Ø
1Ø26       Ø456  ØØØØ                Ø
1Ø27       Ø457  ØØØØ                Ø
1Ø3Ø       Ø46Ø  ØØØØ                Ø
1Ø31       Ø461  ØØØØ                Ø
1Ø32       Ø462  ØØØØ                Ø
1Ø33       Ø463  ØØØØ                Ø
1Ø34       Ø464  ØØØØ                Ø
1Ø35       Ø465  ØØØØ                Ø
1Ø36       Ø466  1437                TTOINT              /TTY INTERRUPTS ENABLED FOR ECHO
1Ø37       Ø467  1616        INTADD, TTIINT              /KBD INTERRUPTS ALWAYS ENABLED
1Ø4Ø                         /
1Ø41                                 EJECT
```

```
              /
              /       PUSHDOWN LISTS
              /
              ISTACK= .-1                 /INTERRUPT STACK
                      INTST4=INTSTK+INTSTK+INTSTK+INTSTK
                      *.+INTST4           /ALLOW SPACE FOR STACK
              /
Ø51Ø   74Ø2   MSTACK, HLT                 /MONITOR STACK
              /
                      *.+MONSTK+MONSTK+MONSTK
              /
              /       GENERAL PURPOSE TABLES
              /
              /       TABLES WHICH MAY CROSS PAGE
              /       BOUNDARIES ARE HERE
              /
              /       TTY INPUT SPECIAL CHARACTERS
              /
Ø525   7575   SPCHAR, -2Ø3                /CONTROL C
Ø526   1766           TTCTLC
Ø527   7566           -212                /LINE FEED
Ø53Ø   1731           TTCRLF
Ø531   7563           -215                /CARRIAGE RETURN
Ø532   1731           TTCRLF
Ø533   7545           -233                /NEW ALTMODE
Ø534   1741           TTALT
Ø535   74Ø3           -375                /ALTMODE
Ø536   1741           TTALT
Ø537   74Ø2           -376                /ESCAPE
Ø54Ø   1741           TTALT
Ø541   74Ø1           -377                /RUBOUT
Ø542   1712           TTRUB
              /
              /       LIST OF I/O ROUTINES BY UNIT CLASS
              /
Ø543   1Ø26   ULIST,  IOHLT               /CLASS Ø INVALID
Ø544   1Ø26           IOHLT               /CLASS 1 INVALID
Ø545   12ØØ           IOLTP               /CLASS 2 LINC TAPE
                      ASMIFZ  RKØ8!RFØ8
                      IOHLT
                      ASMIFN  RKØ8!RFØ8
Ø546   24ØØ           IODISK              /CLASS 3 DISK
Ø547   14ØØ           IOTTY               /CLASS 4 TELETYPE
                      ASMIFZ  MSCDEV
Ø55Ø   1Ø26           IOHLT
                      ASMIFN  MSCDEV
                      IOMISC              /CLASS 5 MISC
                      ASMIFZ  DISPLY
                      IOHLT
                      ASMIFN  DISPLY
Ø551   22Ø1           IODISP              /CLASS 6 DISPLAY
Ø552   1Ø26           IOHLT               /CLASS 7 INVALID
              /
                      EJECT
```

```
                /
                /       PARAMETER LIST FOR RELOADING
                /       JOB CONTROL ON EXIT OR CONTROL C
                /
Ø553  ØØ2Ø      JOBCTL, 2Ø                  /MAY BE MODIFIED BY IPL
Ø554  62Ø1              CDF     Ø
Ø555  34ØØ              JOBENT
Ø556  4ØØØ              74ØØ-JOBENT         /READ TO 74ØØ
Ø557  ØØ1Ø              1Ø                  /MODIFIED BY IPL
Ø56Ø  ØØØØ              Ø
                /       Ø                   /THIS LOC NOT NEEDED
                /
                /
                JOBENT= 34ØØ
                /
                /       I/O CONTROL BLOCK FOR FINAL LOADER
                /
Ø561  ØØ2Ø      LDLIST, 2Ø                  /LOAD UNIT
Ø562  6211              CDF     1Ø          /LAST FIELD
Ø563  74ØØ              74ØØ                /BLOCK ADDRESS
Ø564  Ø4ØØ              4ØØ                 /LENGTH
Ø565  ØØØØ              Ø                   /BLOCK NUMBER
Ø566  ØØØØ              Ø
                /       Ø                   /THIS LOC NOT NEEDED
                /
                /
                        EJECT
```

```
                     /
                             PAGE
                     /
                     /       ROUTINE TO ENABLE AN INTERRUPT
                     /       HANDLER BY INSERTING THE DESIRED
                     /       IOT SKIP AND HANDLER ADDRESS
                     /       IN THE INTERRUPT SCAN CHAIN.
                     /
                     /
      Ø6ØØ  ØØØØ     ISETR,  Ø
      Ø6Ø1  4325             JMS     MNTR
      Ø6Ø2  16ØØ             TAD I   ISETR   /IOT FOR WHICH WE SEARCH
      Ø6Ø3  3ØØ5             DCA     TMP4
      Ø6Ø4  62Ø1             CDF     Ø
      Ø6Ø5  1ØØ5             TAD     TMP4
      Ø6Ø6  7Ø41             CIA
      Ø6Ø7  3ØØ2             DCA     TMP1
      Ø61Ø  4253             JMS     IOTSCN  /SEARCH FOR IT
      Ø611  5214             JMP     .+3     /NO FIND
      Ø612  116Ø             TAD     KNOP    /REPLACE IOT WITH NOP
      Ø613  34Ø3             DCA I   TMP2
      Ø614  4346             JMS     MUDF
      Ø615  22ØØ             ISZ     ISETR
      Ø616  16ØØ             TAD I   ISETR   /HANDLER ADDRESS
      Ø617  7Ø4Ø             CMA             /7777 BECOMES Ø
      Ø62Ø  765Ø             SNA CLA         /DELETE?
      Ø621  5247             JMP     ISXIT   /YES-ALL DONE
      Ø622  62Ø1             CDF     Ø
      Ø623  7332             CLA STL RTR
      Ø624  7Ø1Ø             RAR             /1ØØØ = -NOP
      Ø625  3ØØ2             DCA     TMP1
      Ø626  4253             JMS     IOTSCN
      Ø627  5251             JMP     ISERR   /NO ROOM
      Ø63Ø  1ØØ4             TAD     TMP3
      Ø631  1274             TAD     PINTAD  /ADDRESS HANDLER ADDRESS
      Ø632  3ØØ2             DCA     TMP1
      Ø633  4346             JMS     MUDF
      Ø634  16ØØ             TAD I   ISETR   /NEW HANDLER ADDR
      Ø635  62Ø1             CDF     Ø
      Ø636  34Ø2             DCA I   TMP1    /INTO LIST
      Ø637  73Ø5             TWO
      Ø64Ø  1ØØ3             TAD     TMP2
      Ø641  3ØØ2             DCA     TMP1
      Ø642  73Ø5             TWO
      Ø643  1347             TAD     MUDF+1  /BUILD CIF CDF
      Ø644  34Ø2             DCA I   TMP1    /IN STRING
      Ø645  1ØØ5             TAD     TMP4    /CALLERS IOT
      Ø646  34Ø3             DCA I   TMP2
      Ø647  73Ø5     ISXIT,  TWO             /SKIP TWO ON EXIT
      Ø65Ø  5352             JMP     MXIT
                     /
      Ø651  7325     ISERR,  THREE
      Ø652  5277             JMP     MERR
                     /
                             EJECT
```

```
                        /
                        /
        Ø653  ØØØØ  IOTSCN, Ø
        Ø654  1275          TAD     PINTSC
        Ø655  3ØØ3          DCA     TMP2
        Ø656  1276          TAD     INTLEN
        Ø657  3ØØ4          DCA     TMP3
        Ø66Ø  14Ø3  IOTSLP, TAD I   TMP2        /GET INST FROM CHAIN
        Ø661  1ØØ2          TAD     TMP1        /COMPARE
        Ø662  765Ø          SNA CLA             /EQUAL?
        Ø663  5272          JMP     IOTFND      /YES
        Ø664  73Ø7          FOUR
        Ø665  1ØØ3          TAD     TMP2        /POINT TO NEXT
        Ø666  3ØØ3          DCA     TMP2
        Ø667  2ØØ4          ISZ     TMP3        /CHECK COUNT
        Ø67Ø  526Ø          JMP     IOTSLP
        Ø671  5653          JMP I   IOTSCN
        Ø672  2253  IOTFND, ISZ     IOTSCN
        Ø673  5653          JMP I   IOTSCN
                    /
                    /
        Ø674  Ø466  PINTAD, INTADD-1
        Ø675  Ø4ØØ  PINTSC, INTSCN
        Ø676  777Ø  INTLEN, -INTMAX
                    /
                            EJECT
```

```
                          /
                          /         MONITOR ERROR HANDLER
                          /
                          /         MESSAGES HAVE THE FORM ?N, WHERE N IS:
                          /         0: UNIMPLEMENTED COMMAND
                          /         1: TOO MANY INTERRUPT PUSHES
                          /         2: TOO MANY INTERRUPT POPS
                          /         3: TOO MANY SETINTS
                          /         4: INVALID I/O UNIT
                          /         5: I/O REQUEST ON BUSY CONTROL BLOCK
                          /         6: DISK ERROR
                          /
        0677   1161       MERR,     TAD       CZERO
        0700   6002                 IOF
        0701   3164                 DCA       QT3       /HOLD CODE
        0702   1313                 TAD       MQUEST    /GET ?
        0703   4714                 JMS I     PECHO
        0704   1164                 TAD       QT3       /GET CODE
        0705   4714                 JMS I     PECHO
        0706   6001                 ION
        0707   4315                 JMS       WAITR     /WAIT
        0710   1531                 TTECNT              /FOR COMPLETION
        0711   7402                 HLT                 /THEN STOP
        0712   4425                 JMS I     EXIT      /RETURN TO JCL
                          /
        0713   0277       MQUEST,   277                 /?
        0714   1500       PECHO,    TTECHO
                          /
                          /         SIMPLE WAIT
                          /
        0715   0000       WAITR,    0
        0716   4547                 JMS I     MENTER
        0717   1715                 TAD I     WAITR     /ADDR OF WAIT PARAMETER
        0720   3002                 DCA       TMP1
        0721   1402                 TAD I     TMP1
        0722   7710                 SPA CLA             /IS LIST READY?
        0723   5321                 JMP       .-2       /NO
        0724   5550                 JMP I     MXSKP
                          /
                                    EJECT
```

```
                        /
                        /       MONITOR SERVICE ROUTINES
                        /
        0725    0000    MNTR,   0
        0726    0153            AND     P70     /GET CALLERS IF
        0727    1157            TAD     KCDF    /BUILD CDF
        0730    3347            DCA     MUDF+1
        0731    6214            RDF             /CALLERS DF
        0732    1157            TAD     KCDF    /BUILD CDF
        0733    6201            CDF     0
        0734    3417            DCA I   MSTP    /PUSH ONTO STACK
        0735    7344            MTWO
        0736    1325            TAD     MNTR    /ADDR OF RETURN ADDR
        0737    3002            DCA     MTEMP
        0740    1402            TAD I   MTEMP   /GET RETURN ADDR
        0741    3417            DCA I   MSTP    /PUSH ONTO STACK
        0742    1347            TAD     MUDF+1  /CDF TO USERS IF
        0743    3417            DCA I   MSTP    /ONTO STACK
        0744    4346            JMS     MUDF    /RETURN WITH USERS FIELD
        0745    5725            JMP I   MNTR
                        /
                        /
        0746    0000    MUDF,   0
        0747    6201            CDF             /MODIFIED BY MNTR,
        0750    5746            JMP I   MUDF    /RESTORED BY MXIT
                        /
                        /
        0751    7201    MX1,    ONE             /SETUP FOR SKIP RETURN
        0752    6001    MXIT,   ION
        0753    3002            DCA     MTEMP   /HOLD RETURN ADDR ADJUSTMENT
        0754    6201            CDF     0
        0755    1156            TAD     M4
        0756    1017            TAD     MSTP
        0757    3017            DCA     MSTP    /POP BACK TO PREVIOUS CDF
        0760    1417            TAD I   MSTP
        0761    3347            DCA     MUDF+1  /RESTORE NEXT LEVEL CALLER FIELD
        0762    1417            TAD I   MSTP    /GET THIS LEVEL DF
        0763    3372            DCA     MXDF    /EXIT DATA FIELD
        0764    1002            TAD     MTEMP   /GET RETURN ADDR MODIFICATION
        0765    1417            TAD I   MSTP    /ADD TO RETURN ADDR
        0766    3002            DCA     MTEMP   /SET RETURN ADDR
        0767    7201            ONE
        0770    1417            TAD I   MSTP    /CONVERT CALLER CDF TO CIF
        0771    3376            DCA     MXIF
        0772    6201    MXDF,   CDF             /RESTORE USER DF
        0773    7346            MTHREE
        0774    1017            TAD     MSTP    /RESTORE STACK POINTER
        0775    3017            DCA     MSTP
        0776    6202    MXIF,   CIF             /RESTORE INST FIELD
        0777    5402            JMP I   MTEMP   /RETURN TO HIM
                        /
                        MTEMP=  TMP1
                        /
                                EJECT
```

```
            /
                    PAGE
            /
            /       GENERAL I/O ROUTINES
            /
1ØØØ  ØØØØ  READR,  Ø                   /INPUT ENTRY POINT
1ØØ1  4547          JMS I   MENTER
1ØØ2  521Ø          JMP     IOCOM       /BEGIN COMMON
1ØØ3  ØØØØ  WRITR,  Ø
1ØØ4  4547          JMS I   MENTER
1ØØ5  12Ø3          TAD     WRITR
1ØØ6  32ØØ          DCA     READR
1ØØ7  724Ø          MONE
1Ø1Ø  3ØØ7  IOCOM,  DCA     RWSW        /=Ø IF READ, -1 IF WRITE
1Ø11  16ØØ          TAD I   READR
1Ø12  3ØØ6          DCA     TMP5        /PARAMETER LIST ADDR
1Ø13  14Ø6          TAD I   TMP5
1Ø14  Ø153          AND     P7Ø         /GET UNIT CLASS
1Ø15  711Ø          CLL RAR
1Ø16  7Ø12          RTR                 /INDEX THE LIST
1Ø17  1225          TAD     PULIST      /OF ROUTINES
1Ø2Ø  3ØØ3          DCA     TMP2
1Ø21  62Ø1          CDF     Ø
1Ø22  14Ø3          TAD I   TMP2        /ROUTINE ADDRESS
1Ø23  3ØØ3          DCA     TMP2
1Ø24  54Ø3          JMP I   TMP2
            /
1Ø25  Ø543  PULIST, ULIST
1Ø26  73Ø7  IOHLT,  FOUR
1Ø27  5552          JMP I   MERROR
            /
                    EJECT
```

```
                  /       ENQUEUE A PARAMETER LIS
                  /       RETURNS WITH DATA FIELD SET TO USER
                  /       AND INTERRUPTS OFF
                  /
                  /       JMS I   NQUEUE
                  /       Q POINTER
                  /       JMP             /OTHERS ARE IN LIST
                  /                       /THIS IS THE FIRST
1Ø3Ø  ØØØØ  ENQ,          Ø
1Ø31  163Ø                TAD I   ENQ     /POINTER TO DEVICE Q POINTER
1Ø32  6ØØ2                IOF
1Ø33  3163                DCA     QT2     /HOLD FOR ADDRESSING
1Ø34  223Ø                ISZ     ENQ     /BUMP RETURN
1Ø35  1776                TAD I   PMUDF1  /PICK UP USER CDF
1Ø36  3362                DCA     QMLOOP
1Ø37  1563                TAD I   QT2     /CDF TO FIRST LIST ON CHAIN
1Ø4Ø  745Ø                SNA             /IS CHAIN EMPTY?
1Ø41  5265                JMP     NQFRST  /YES
1Ø42  3247  NQSRCH,       DCA     NQCDF   /NO-POINT TO NEXT ENTRY
1Ø43  2163                ISZ     QT2
1Ø44  1155                TAD     P5
1Ø45  1563                TAD I   QT2     /ADDRESS NEXT CHAIN POINTER
1Ø46  3163                DCA     QT2
1Ø47  62Ø1  NQCDF,        CDF
1Ø5Ø  1563                TAD I   QT2     /PICK UP NEXT CDF
1Ø51  744Ø                SZA             /END OF CHAIN?
1Ø52  5242                JMP     NQSRCH  /NO-FIND NEXT ENTRY
1Ø53  1155                TAD     P5      /BUMP PARAM ADDR BY 5
1Ø54  4272                JMS     NQCHN
1Ø55  4551  NQXIT,        JMS I   SETUDF  /POINT TO CALLER FIELD
1Ø56  733Ø                STL CLA RAR     /4ØØØ IN AC
1Ø57  14Ø6                TAD I   TMP5    /SET BUSY
1Ø6Ø  75ØØ                SMA             /DID BIT SET?
1Ø61  5352                JMP     QERR    /NO
1Ø62  34Ø6                DCA I   TMP5
1Ø63  3562                DCA I   QT1     /CLEAR HIS CHAIN WORD
1Ø64  563Ø                JMP I   ENQ     /RETURN TO DEVICE ROUTINE
                  /
                  /       COME HERE IF THIS IS FIRST ENTRY IN QUEUE
                  /
1Ø65  4272  NQFRST,       JMS     NQCHN   /SETUP CHAIN
1Ø66  4355                JMS     QMOVE   /MOVE PARAMS DOWN
1Ø67  2162                ISZ     QT1     /BUMP POINTER TO CHAIN WORD
1Ø7Ø  223Ø                ISZ     ENQ     /TAKE SECOND RETURN
1Ø71  5255                JMP     NQXIT
                  /
                  /       ADD THIS PARAM LIST TO CHAIN
                  /
1Ø72  ØØØØ  NQCHN,        Ø
1Ø73  1ØØ6                TAD     TMP5    /PARAM ADDR,,,
1Ø74  3162                DCA     QT1     /,,,TO MOVE POINTER
1Ø75  1362                TAD     QMLOOP  /USER CDF,,,
1Ø76  3563                DCA I   QT2     /,,,TO LIST
1Ø77  2163                ISZ     QT2
11ØØ  1ØØ6                TAD     TMP5    /ALSO PARAM ADDR
11Ø1  3563                DCA I   QT2
11Ø2  5672                JMP I   NQCHN
                  /
                          EJECT
```

```
                /
                /
                /       DEQUEUE A PARAMETER LIST
                /       MUST BE CALLED WITH INTERRUPTS OFF
                /
                /       JMS I   DQUEUE
                /       Q POINTER
                /       JMP             /NOTHING LEFT IN LIST
                /                       /Q POINTER IS SET TO NEXT
                /
1103  ØØØØ      DEQ,    Ø
1104  62Ø1              CDF     Ø
1105  1/Ø3              TAD I   DEQ     /Q POINTER ADDR
11Ø6  3162              DCA     QT1
11Ø7  1162              TAD     QT1     /CDF IN QP
111Ø  7ØØ1              IAC
1111  3163              DCA     QT2     /PARAM ADDR
1112  23Ø3              ISZ     DEQ     /ADJUST RETURN ADDR
1113  1562              TAD I   QT1
1114  3317              DCA     DQCDF
1115  1563              TAD I   QT2     /POINTER TO PARAM LIST
1116  3164              DCA     QT3
1117  62Ø1      DQCDF,  CDF             /MODIFIED ABOVE
112Ø  735Ø              STA CLL RAR     /3777 IN AC
1121  Ø564              AND I   QT3     /GET ALL BUT Ø
1122  3564              DCA I   QT3     /STORE WITHOUT BUSY BIT
1123  1155              TAD     P5
1124  1164              TAD     QT3     /GET CHAIN POINTER
1125  3164              DCA     QT3     /ADDRESS
1126  1564              TAD I   QT3
1127  62Ø1              CDF     Ø
113Ø  3562              DCA I   QT1     /STORE IN QP
1131  1562              TAD I   QT1     /IS THERE ANOTHER?
1132  745Ø              SNA
1133  57Ø3              JMP I   DEQ     /NO-RETURN NOTHING
1134  3362              DCA     QMLOOP  /ELSE SET CDF TO NEXT LIST
1135  23Ø3              ISZ     DEQ     /AND BUMP RETURN
1136  1317              TAD     DQCDF   /CDF TO OLD LIST
1137  334Ø              DCA     .+1
114Ø  62Ø1              CDF             /OLD PARAMETER FIELD
1141  3564              DCA I   QT3     /CLEAR CDF IN LIST
1142  2164              ISZ     QT3
1143  1564              TAD I   QT3
1144  62Ø1              CDF     Ø
1145  3162              DCA     QT1     /PARAMETER ADDR
1146  1162              TAD     QT1
1147  3563              DCA I   QT2     /POINT TO LIST
115Ø  4355              JMS     QMOVE   /MOVE INTO Q POINTER
1151  57Ø3              JMP I   DEQ     /RETURN
                /
1152  73Ø7      QERR,   FOUR
1153  7ØØ1              IAC
1154  5552              JMP I   MERROR
                /
                        EJECT
```

```
            /
            /
1155  0000  QMOVE,  0
1156  1156          TAD     M4
1157  3164          DCA     QT3
1160  7346          MTHREE
1161  3165          DCA     QT4     /SPECIAL COUNT
1162  6201  QMLOOP, CDF             /ADDR NEXT PARAM
1163  2162          ISZ     QT1
1164  2163          ISZ     QT2
1165  1562          TAD I   QT1     /GET A WORD
1166  6201          CDF     0
1167  2165          ISZ     QT4     /IS THIS LENGTH?
1170  7410          SKP             /NO
1171  7041          CIA             /YES, MAKE NEG
1172  3563          DCA I   QT2
1173  2164          ISZ     QT3
1174  5362          JMP     QMLOOP
1175  5755          JMP I   QMOVE
            /
            /
1176  0747  PMUDF1, MUDF+1          /ADDR OF USER CDF
            /
                    EJECT
```

```
                        /
                                PAGE
                        /
                        /       LINC TAPE I/O ROUTINE
                        /
                        /       MUST BE IN SEGMENT Ø
                        /
        12ØØ  4545      IOLTP,  JMS I   NQUEUE
        12Ø1  1214              LTPQP
        12Ø2  5212              JMP     LTQED   /OTHERS IN Q
                        /
                        /       START THIS OPERATION
                        /
        12Ø3  12Ø4              TAD     .+1     /GET LINC INST
        12Ø4  6141              LINC
                                LMODE
        12Ø5  445Ø              STC     LTLINC  /ENABLE SCANNER
        12Ø6  ØØØ2              PDP
                                PMODE
        12Ø7  4222              JMS     SETRW   /SET READ/WRITE
        121Ø  4233              JMS     LTPDO   /START THE OP
        1211  555Ø              JMP I   MXSKP   /RETURN
                        /
        1212  4222      LTQED,  JMS     SETRW   /SET READ/WRITE
        1213  555Ø              JMP I   MXSKP   /RETURN
                        /
                        /
        1214  ØØØØ      LTPQP,  Ø               /CDF TO CURRENT LIST
        1215  ØØØØ              Ø               /LIST ADDR
        1216  ØØØØ              Ø               /BUFFER CDF
        1217  ØØØØ              Ø               /BUFFER ADDR
        122Ø  ØØØØ              Ø               /REMAINING COUNT
        1221  ØØØØ              Ø               /CURRENT BLOCK
                        /
                        /       SET THE READ/WRITE BIT (BIT 1 OF UNIT)
                        /       IN THIS PARAMETER LIST
                        /
        1222  ØØØØ      SETRW,  Ø
        1223  14Ø6              TAD I   TMP5    /GET UNIT
        1224  71Ø6              CLL RTL
        1225  712Ø              STL             /LINK = 1 FOR WRITE
        1226  2ØØ7              ISZ     RWSW    /SKIP IF WRITE
        1227  71ØØ              CLL             /CLEAR LINK FOR READ
        123Ø  7Ø12              RTR             /RESTORE UNIT
                                                /READ/WRITE IN BIT 1
        1231  34Ø6              DCA I   TMP5
        1232  5622              JMP I   SETRW   /RETURN TO CALLER
                        /
                                EJECT
```

```
                  /
                  /       THIS DOES THE ACTUAL OPERATION
                  /
1233  0000  LTPDO,  0
1234  1221          TAD     LTPQP+5 /BLOCK NUMBER
1235  3272          DCA     LTBLK   /MOVE TO OP
1236  1615          TAD I   LTPQP+1
1237  7006          RTL             /READ/WRITE BIT TO LINK
1240  7630          SZL CLA         /SKIP IF READ
1241  7307          FOUR            /ELSE MAKE WRITE
1242  1332          TAD     LTRDE   /BUILD BASIC INSTRUCTION
1243  3271          DCA     LTOP
1244  7327          SIX             /MASK FOR UNIT
1245  7001          IAC             /7 IN AC
1246  0615          AND I   LTPQP+1 /GET UNIT
1247  7110          CLL RAR         /LOW ORDER TO LINK
1250  3331          DCA     LT1     /SAVE HIGH ORDER
1251  7006          RTL
1252  7006          RTL             /LOW ORDER TO BIT 8
1253  1271          TAD     LTOP    /COMBINE UNIT
1254  3271          DCA     LTOP    /WITH BASIC OP
1255  1216          TAD     LTPQP+2 /BUFFER FIELD
1256  0153          AND     P70
1257  7106          CLL RTL
1260  7006          RTL
1261  7006          RTL             /FIELD TO BITS 0-2
1262  1331          TAD     LT1     /HI ORDER UNIT
1263  1330          TAD     LTPXOB  /OTHERS
1264  6141          LINC
                    LMODE
1265  0001          AXO             /SET EXT OPS
1266  1000          LDA
1267  1217          LTPQP+3         /BUFFER ADDR
1270  0023          TMA             /SET ADDRESS
1271  0702  LTOP,   RDE             /READ OR WRITE
1272  0000  LTBLK,  0               /THIS BLOCK
1273  0002          PDP
                    PMODE
1274  5633          JMP I   LTPDO   /OPERATION STARTED
            /
            /       GOOD PROJECT FOR SOMEDAY:
            /       MAKE WRITE CHECKING OPTIONAL
            /
                    EJECT
```

```
                     /
                     /        LINC TAPE INTERRUPT HANDLER
                     /
                     /        MUST BE IN SEGMENT Ø
                     /        ENTERED BY LINC MODE JMP
                     /        RETURN TO INTEX
                     /
1275  ØØØ2   LTPINT, ØØØ2                /ENTER PDP MODE
1276  73ØØ           CLA CLL
1277  1334           TAD      LTBLEN   /BLOCK LEN
13ØØ  122Ø           TAD      LTPQP+4  /PLUS COUNT
13Ø1  322Ø           DCA      LTPQP+4  /UPDATE COUNT
13Ø2  763Ø           SƵL CLA           /MORE TO GO?
13Ø3  5315           JMP      LTPNXT   /NO-GET NEXT FROM Q
13Ø4  1334           TAD      LTBLEN   /YES-BUMP
13Ø5  1217           TAD      LTPQP+3  /BUFFER ADDR
13Ø6  3217           DCA      LTPQP+3
13Ø7  2221           ISƵ      LTPQP+5  /BUMP BLOCK NO
131Ø  1214   LTPCDF, TAD      LTPQP    /GET PARAM CDF
1311  3312           DCA      .+1
1312  62Ø1           CDF               /MODIFIED TO ADDRESS PARAM LIST
1313  4233           JMS      LTPDO    /DO THE OPERATION
1314  5733           JMP I    LTEXIT
                     /
                     /        THIS OPERATION IS COMPLETE
                     /        GET THE NEXT
                     /
1315  4546   LTPNXT, JMS I    DQUEUE
1316  1214           LTPQP
1317  5321           JMP      LTICLR   /NO MORE ON Q
132Ø  531Ø           JMP      LTPCDF   /DO IT
                     /
                     /        NO MORE ON Q, DISABLE INTERRUPTS
                     /
1321  6141   LTICLR, LINC
                     LMODE
1322  1ØØØ           LDA               /REMOVE LTPINT
1323  Ø454           LTNONE            /FROM INTERRUPT
1324  445Ø           STC      LTLINC   /SCAN CHAIN
1325  ØØØ1           AXO               /CLEAR XOB TO PREVENT INTERRUPTS
1326  ØØØ2           PDP
                     PMODE
1327  5733           JMP I    LTEXIT   /RETURN
                     /
133Ø  Ø13Ø   LTPXOB, 13Ø               /ENABLE INTERRUPTS,
                                       /EXTENDED ADDR,
                                       /AND NO PAUSE
1331  ØØØØ   LT1,    Ø
1332  Ø7Ø2   LTRDE,  7Ø2
1333  Ø267   LTEXIT, INTEX
1334  Ø4ØØ   LTBLEN, 4ØØ
                     /
                              EJECT
```

```
                  /
                  /       JOB CONTROL BOOTSTRAP
                  /
1335  ØØØØ  JCBOOT, Ø
1336  6141          LINC
                    LMODE
1337  1Ø2Ø          LDA I
134Ø  ØØ2Ø          2Ø
1341  ØØØ4          ESF                 /IO PRESET TO CLEAR ALL FLAGS
1342  1ØØØ          LDA
1343  Ø454          LTNONE              /CLEAR LTP INTERRUPT
1344  445Ø          STC       LTLINC    /SCAN CHAIN ENTRY
1345  ØØØ2          PDP
                    PMODE
1346  1362          TAD       RCCP      /RESTORE CONTROL
1347  3177          DCA       TTPCTL    /CHAR HANDLER
135Ø  62Ø1          CDF       Ø
1351  3214          DCA       LTPQP
                    ASMIFN    RKØ8
1352  3761          DCA I     PQP       /CLEAR Q ON SYSTEM DEVICE
1353  4421          JMS I     READ      /GET JOB CONTROL
1354  Ø553          JOBCTL
1355  4423          JMS I     WAIT      /WAIT FOR IT
1356  Ø553          JOBCTL
1357  576Ø          JMP I     .+1       /ENTER JOB CONTROL
136Ø  34ØØ          JOBENT
                  /
                    ASMIFN    RKØ8
1361  2413  PQP,    DKQP
1362  1637  RCCP,   TTINRM-1            /CONSTANT FOR RESTORING
                  /                     CONTROL CHAR HANDLER
                  /
                  /
                  /       END OF LOADER
                  /
                    *.&76ØØ+171         /FOLLOWING AT END OF PAGE
                  /
                  /       READS THE LAST BLOCK OF A BINARY FILE
                  /       OVERLAYING THE LOADER, THEN SETS THE MODE
                  /       AND JUMPS TO THE PROGRAM STARTING ADDRESS.
                  /
1371  4421  LDLAST, JMS I     READ      /READ LAST BLOCK
1372  Ø561          LDLIST
1373  4423          JMS I     WAIT      /WAIT COMPLETION
1374  Ø561          LDLIST
1375  62Ø2  LDSTRT, CIF                 /"CIF X" OR "LINC"
1376  5777          JMP I     .+1       /"JMP" OR "LIF"
1377  ØØØØ          Ø                   /ADDR OR "JMP"
                  /
                    ASMIFN    .&177
                  ERROR               /ADJUST ORG ABOVE TO PUT THIS
                  /       AT PAGE BOUNDARY.
                  /
                    EJECT
                    /
                            PAGE
                    /
                    /       TTY I/O ROUTINE
                    /
                    /       KEYBOARD AND PRINTER INTERRUPT
                    /       HANDLERS ARE ALWAYS ENABLED
                    /
      1400  2007    IOTTY,  ISZ     RWSW    /0=READ, -1=WRITE
      1401  5630            JMP I   TTIQER  /INPUT
                    /
                    /       QUEUE OUTPUT REQUEST
                    /
      1402  4545    TTOQ,   JMS I   NQUEUE
      1403  1421            TTOQP
      1404  5550            JMP I   MXSKP   /RETURN
      1405  1223            TAD     TTOQP+2 /BUFFER CDF
      1406  3232            DCA     TTOCDF
      1407  2227            ISZ     TTOFLG  /IS ANYTHING RUNNING?
      1410  5215            JMP     TTCLUG  /YES - WAIT TILL HES DONE
      1411  7240            MONE
      1412  3227            DCA     TTOFLG  /NO - RESET FLAG
      1413  4231            JMS     TTOPUT  /START THIS
      1414  5550            JMP I   MXSKP
                    /
                    /       IN THE UNUSUAL CASE THAT AN OUTPUT REQUEST
                    /       COMES WHILE AN INPUT CHARACTER IS BEING ECHOED,
                    /       THIS FIXES UP THE COUNT.
                    /
      1415  7240    TTCLUG, MONE
      1416  1225            TAD     TTOQP+4 /ADJUST COUNT IN
      1417  3225            DCA     TTOQP+4 /SPECIAL CASE
      1420  5550            JMP I   MXSKP
                    /
                    /
      1421  0000    TTOQP,  0               /CDF TO PARAMS IF Q ACTIVE
      1422  0000            0               /PARAM LIST ADDR
      1423  0000            0               /BUFFER CDF
      1424  0000            0               /CURRENT BUFFER LOCATION
      1425  0000            0               /REMAINING COUNT
      1426  0000            0               /IGNORED
                    /
      1427  7777    TTOFLG, -1              /-1 IF NO PRINTING ACTIVE NOW
      1430  1600    TTIQER, TTIQ
                    /
                    /       PRINT NEXT CHAR IN BUFFER
                    /
      1431  0000    TTOPUT, 0
      1432  6201    TTOCDF, CDF
      1433  1624            TAD I   TTOQP+3 /GET NEXT CHAR
      1434  2224            ISZ     TTOQP+3 /BUMP ADDR
      1435  4300            JMS     TTECHO  /OUTPUT CHAR
      1436  5631            JMP I   TTOPUT
                    /
                            EJECT
```

```
            /
            /        TELEPRINTER INTERRUPT HANDLER
            /
            /        PENDING ECHO REQUESTS ARE HONORED BEFORE NORMAL
            /        OUTPUT. THIS MAY BE CHANGED BY CHANGING THE
            /        ORDER OF THE TESTS ON TTECNT AND TTOQP, BELOW.
            /
1437  ØØØØ  TTOINT,  Ø
1440  72ØØ           CLA
1441  1331           TAD     TTECNT  /ECHO COUNT
1442  771Ø           SPA CLA         /ANY PENDING?
1443  5264           JMP     TTEOUT  /YES - ECHO TAKES PRIORITY
1444  1221           TAD     TTOQP   /NO - LOOK AT Q
1445  765Ø           SNA CLA         /IS THERE AN OUTPUT REQUEST?
1446  526Ø           JMP     TTOCLR  /NO - CLEAR FLAG AND QUIT
1447  2225           ISZ     TTOQP+4 /CHECK REMAINING COUNT
145Ø  5256           JMP     TTONXT  /MORE TO GO
1451  4546           JMS I   DQUEUE  /END OF REQUEST - GET NEXT
1452  1421           TTOQP
1453  526Ø           JMP     TTOCLR  /NO MORE - CLEAR FLAG
1454  1223           TAD     TTOQP+2 /GET NEXT BUFFER CDF
1455  3232           DCA     TTOCDF
1456  4231  TTONXT,  JMS     TTOPUT  /PUT NEXT OUT
1457  5637           JMP I   TTOINT  /CONTINUE UNTIL NEXT INTERRUPT
            /
            /        THERE IS NOTHING MORE TO BE PRINTED,
            /        WE MUST CLEAR FLAG TO PREVENT INTERRUPT LOOPS.
            /
146Ø  724Ø  TTOCLR,  MONE
1461  3227           DCA     TTOFLG  /READY FOR ANYTHING
1462  6Ø42           TCF             /CLEAR FLAG
1463  5637           JMP I   TTOINT  /AND EXIT
            /
            /        ECHO HANDLER
            /
1464  1732  TTEOUT,  TAD I   TTEBOP  /NEXT ECHO CHAR
1465  6Ø46           TLS             /SEND IT
1466  72ØØ           CLA             /FORGET IT
1467  2331           ISZ     TTECNT  /REMAINING ECHO COUNT
147Ø  741Ø           SKP             /MORE TO GO
1471  5637           JMP I   TTOINT  /END OF ECHO - QUIT
1472  72Ø1           ONE
1473  1332           TAD     TTEBOP  /INCR POINTER
1474  Ø334           AND     TTEBLN  /FORCE WRAP-AROUND
1475  1333           TAD     TTEBUF
1476  3332           DCA     TTEBOP  /NEW POINTER
1477  5637           JMP I   TTOINT
            /
                     EJECT
            /
            /       ECHO CHAR IN AC, BUFFERED
            /
15ØØ  ØØØØ  TTECHO, Ø
15Ø1  62Ø1          CDF     Ø
15Ø2  3163          DCA     TTECHR
15Ø3  1334          TAD     TTEBLN    /BUFFER SIZE
15Ø4  1331          TAD     TTECNT    /COMPARE BUFFER USE
15Ø5  771Ø          SPA CLA           /BUFFER FULL?
15Ø6  57ØØ          JMP I   TTECHO    /YES-DO NOT ECHO
15Ø7  2227          ISZ     TTOFLG    /IS ANYTHING GOING ON?
151Ø  5315          JMP     TTEQ      /YES - Q THIS
1511  1163          TAD     TTECHR    /NO - PUT OUT NOW
1512  6Ø46          TLS
1513  72ØØ          CLA
1514  57ØØ          JMP I   TTECHO    /RESUME
            /
            /       SOMETHING IS PRINTING --
            /       WE MUST PUT THIS CHAR IN THE RING BUFFER,
            /
1515  1331  TTEQ,   TAD     TTECNT
1516  7Ø41          CIA               /POS BUFFER LOAD
1517  1332          TAD     TTEBOP    /PLUS OUTPUT ADDR
152Ø  Ø334          AND     TTEBLN    /WRAP AROUND
1521  1333          TAD     TTEBUF
1522  3162          DCA     TTETMP
1523  1163          TAD     TTECHR
1524  3562          DCA I   TTETMP
1525  724Ø          MONE
1526  1331          TAD     TTECNT    /DECR COUNT
1527  3331          DCA     TTECNT
153Ø  57ØØ          JMP I   TTECHO
            /
            /
            TTETMP= QT1
            TTECHR= QT2
1531  ØØØØ  TTECNT, Ø
1532  154Ø  TTEBOP, TTEBFF            /VARIABLE-CURRENT OUTPUT POINTER
1533  154Ø  TTEBUF, TTEBFF            /CONSTANT-START OF BUFFER
1534  ØØØ3  TTEBLN, TTBLEN-1
            /
            /       BECAUSE OF THE RING BUFFERING SCHEME USED,
            /       THE ECHO BUFFER MUST BE A POWER OF TWO
            /       WORDS LONG, AND MUST START AT A LOCATION
            /       A MULTIPLE OF ITS LENGTH, CAVEAT EMPTOR,
            /
            TTMBLN= -TTBLEN           /USEFUL VALUE
                    *,+TTBLEN-1&TTMBLN
154Ø  ØØØØ  TTEBFF, Ø
                    *TTEBFF+TTBLEN
            /
            /       THIS SPACE SHOULD BE USED FOR A
            /       ROUTINE TO KEEP TRACK OF TAB POSITIONS
            /
                    EJECT
```

```
                /
                        PAGE
                /
                /       QUEUE INPUT REQUEST
                /
1600  4545      TTIQ,   JMS I   NQUEUE
1601  1611              TTIQP           /INPUT Q POINTER
1602  5550              JMP I   MXSKP
1603  1213              TAD     TTIQP+2 /BUFFER CDF
1604  3305              DCA     TTICDF
1605  1214              TAD     TTIQP+3 /BUFFER ADDR
1606  7041              CIA             /COMPLEMENTED IS
1607  3346              DCA     TTRLIM  /RUBOUT LIMIT
1610  5550              JMP I   MXSKP   /READY TO GO
                /
                /
1611  ØØØØ      TTIQP,  Ø               /CDF IF CHAIN GOING
1612  ØØØØ              Ø               /PARAM LIST ADDR
1613  ØØØØ              Ø               /BUFFER CDF
1614  ØØØØ              Ø               /CURRENT LOCATION IN BUFFER
1615  ØØØØ              Ø               /REMAINING COUNT
1616  ØØØØ              Ø               /IGNORED
                /
                /       KEYBOARD INTERRUPT HANDLER
                /
                TTIINT= .-1             /USE EXTRA LOC
1617  6036              KRB             /GET THE CHAR
1620  Ø274              AND     TT177   /STRIP PARITY
1621  1275              TAD     TT2ØØ   /FORCE ON
1622  3232              DCA     TTICHR  /HOLD IT
1623  4427              JMS I   INTPSH  /SAVE INTERRUPT STATUS
1624  6ØØ1              ION             /ALLOW INTERRUPTS WHILE WE WORK
                /
                /       IS IT CONTROL?
                /
1625  1232              TAD     TTICHR  /GET CHAR
1626  1276              TAD     TTM24Ø  /COMPARE BLANK
1627  77ØØ              SMA CLA         /IS IT CONTROL?
163Ø  524Ø              JMP     TTINRM  /NO, NORMAL
1631  4577              JMS I   TTPCTL  /GO TO USERS HANDLER
1632  ØØØØ      TTICHR, Ø               /WITH THIS CHAR
1633  524Ø              JMP     TTINRM  /PROCESS NORMALLY
1634  4355              JMS     TTECTL  /ELSE ECHO AS CONTROL
1635  6ØØ2              IOF
1636  5272              JMP     TTIXIT  /AND QUIT
                /
                        EJECT
```

```
            /
            /       IS THIS A SPECIAL CHARACTER?
            /
1637  ØØØØ          Ø                /DUMMY SUBR LOC
164Ø  1277  TTINRM, TAD     TTISPC   /POINTER TO SPECIAL CHAR TABLE
1641  33Ø3          DCA     TTITMP
1642  1232  TTISCN, TAD     TTICHR   /INPUT CHAR
1643  17Ø3          TAD I   TTITMP   /COMPARE SPECIAL
1644  23Ø3          ISZ     TTITMP
1645  745Ø          SNA              /EQUAL?
1646  53ØØ          JMP     TTIFSP   /YES-GO TO THAT
1647  23Ø3          ISZ     TTITMP   /POINT TO NEXT CHAR
165Ø  774Ø  TTM4Ø,  SMA SZA CLA      /TOO FAR FOR POSSIBLE EQUAL?
1651  5242          JMP     TTISCN   /NO-TRY NEXT
            /
            /       NORMAL CHARACTER
            /
1652  43Ø3          JMS     TTIACT   /VERIFY ACTIVITY
1653  1232          TAD     TTICHR   /GET THE CHAR
1654  3614          DCA I   TTIQP+3  /PUT IN THE BUFFER
1655  2214          ISZ     TTIQP+3
1656  1232          TAD     TTICHR
1657  4747          JMS I   TTECP    /ECHO THE CHAR
166Ø  2215          ISZ     TTIQP+4  /ANY MORE?
1661  5272          JMP     TTIXIT   /YES-WAIT FOR NEXT
            /
            /       THIS REQUEST IS SATISFIED,
            /       LOOK FOR NEXT REQUEST ON QUEUE,
            /
1662  4546  TTINXT, JMS I   DQUEUE   /GET NEXT LIST OFF QUEUE
1663  1611          TTIQP
1664  5272          JMP     TTIXIT   /NONE-QUIT NOW
1665  1213          TAD     TTIQP+2  /GET BUFFER CDF
1666  33Ø5          DCA     TTICDF
1667  1214          TAD     TTIQP+3  /BUFFER START ADDR
167Ø  7Ø41          CIA              /COMPLEMENTED
1671  3346          DCA     TTRLIM   /IS RUBOUT LIMIT
1672  443Ø  TTIXIT, JMS I   INTPOP
1673  5616          JMP I   TTIINT
            /
1674  Ø177  TT177,  177
1675  Ø2ØØ  TT2ØØ,  2ØØ
1676  754Ø  TTM24Ø, -24Ø
1677  Ø525  TTISPC, SPCHAR           /LIST OF SPECIAL CHARACTERS
            /
            /
17ØØ  17Ø3  TTIFSP, TAD I   TTITMP   /ADDR OF ROUTINE
17Ø1  33Ø3          DCA     TTITMP
17Ø2  57Ø3          JMP I   TTITMP   /GO HANDLE SPECIAL CHARACTER
            /
                    EJECT
                    /
                    /       VERIFY THAT INPUT IS ACTIVE,
                    /       AND SET BUFFER DATA FIELD
                    /
                    TTITMP= .                   /USE ENTRY FOR TEMPORARY
1703  ØØØØ  TTIACT, Ø
1704  1211          TAD     TTIQP   /Q ACTIVITY INDICATOR
1705  62Ø1  TTICDF, CDF             /SET BUFFR FIELD
17Ø6  6ØØ2          IOF             /STOP INTERRUPTS
17Ø7  764Ø  TTM14Ø, SZA CLA         /IS INPUT UP?
171Ø  57Ø3          JMP I   TTIACT  /YES
1711  5272          JMP     TTIXIT  /NO
                    /
                    /       HANDLERS FOR SPECIAL CHARACTERS
                    /
                    /       CHAR IS 377, RUBOUT
                    /
1712  43Ø3  TTRUB,  JMS     TTIACT  /INPUT ACTIVE?
1713  1214          TAD     TTIQP+3 /CHECK FOR BEGIN
1714  1346          TAD     TTRLIM  /OF BUFFER
1715  765Ø          SNA CLA         /IS IT AN ATTEMPT TO
1716  5272          JMP     TTIXIT  /RUB NON-EXISTANT CHARACTER
1717  724Ø          MONE            /OK-DO IT
172Ø  1214          TAD     TTIQP+3 /DECREMENT ADDR
1721  3214          DCA     TTIQP+3
1722  3614          DCA I   TTIQP+3 /CLEAR LAST CHAR
1723  135Ø          TAD     BACKSL  /GET BACKSLASH
1724  4747          JMS I   TTECP   /ECHO IT
1725  724Ø          MONE
1726  1215          TAD     TTIQP+4 /INCR REMAINING COUNT
1727  3215          DCA     TTIQP+4
173Ø  5272          JMP     TTIXIT  /SO EASY
                    /
                    /       CHAR IS 212 OR 215, LINE FEED OR CARRIAGE RETURN
                    /
1731  43Ø3  TTCRLF, JMS     TTIACT  /INPUT ACTIVE?
1732  1232          TAD     TTICHR  /GET CHAR
1733  3614          DCA I   TTIQP+3 /INTO BUFFER
1734  1351          TAD     CARRET  /CARRIAGE RETURN
1735  4747          JMS I   TTECP   /ECHO NEW LINE
1736  1352          TAD     LINFED
1737  4747  TTSXIT, JMS I   TTECP
174Ø  5262          JMP     TTINXT  /GET NXT REQUEST
                    /
                    /       CHAR IS 233, 375, OR 376, ALTMODE
                    /
1741  43Ø3  TTALT,  JMS     TTIACT  /INPUT ACTIVE?
1742  1354          TAD     ESCAPE  /GIVE USER 233
1743  3614          DCA I   TTIQP+3
1744  1353          TAD     DOLLAR  /ECHO DOLLAR
1745  5337          JMP     TTSXIT  /SPECIAL EXIT
                    /
1746  ØØØØ  TTRLIM, Ø
1747  15ØØ  TTECP,  TTECHO
175Ø  Ø334  BACKSL, 334
1751  Ø215  CARRET, 215
1752  Ø212  LINFED, 212
1753  Ø244  DOLLAR, 244
1754  Ø233  ESCAPE, 233
```

```
                     /
                     /       ECHO CONTROL CHARACTER
                     /       AS UP ARROW, CHAR
                     /
      1755  ØØØØ   TTECTL, Ø
      1756  1364           TAD      UPARRW
      1757  4747           JMS I    TTECP
      176Ø  1232           TAD      TTICHR
      1761  1365           TAD      TT1ØØ    /CONVERT TO NON-CONTROL CHAR
      1762  4747           JMS I    TTECP
      1763  5755           JMP I    TTECTL
                     /
      1764  Ø336   UPARRW, 336
      1765  Ø1ØØ   TT1ØØ,  1ØØ
                     /
                     /
                     /       CONTROL C, EXIT ON FIRST OCCURANCE
                     /       IGNORE MULTIPLE OCCURANCES,
                     /
      1766  2371   TTCTLC, ISZ      CTLCSW   /TEST SWITCH
      1767  5272           JMP      TTIXIT   /HOLD ON, DENNY
      177Ø  4425           JMS I    EXIT
                     /
      1771  7777   CTLCSW, -1
                     /
                     /
                           ASMIFM   2ØØØ-.
                   WARNING          /DISPLAY ROUTINES OVERLAY THIS CODE
                     /
                     /
                     /       MAØ2
                     /
                           CHAIN    "MBØ2"
```

```
        *20
        /
        /       OPTIONAL LDP I/O ROUTINES
        /
                EJECT
```

```
                         ASMIFZ  DIAL
                         SEGMNT  5
                         ASMIFN  DIAL
                         SEGMNT  1
                   /
                         ASMIFZ  GRIDS!DISPLY
                         ASMSKP  227-14
                   /
                   /     4*6 GRIDS FOR 1968 ASCII CHARS
                   /
ØØØØ  ØØØØ               Ø                 /UNUSED
ØØØ1  ØØØØ  HORZ,        Ø
ØØØ2  ØØØØ  DIGP,        Ø
ØØØ3  4437               4437              /Ø1:A
ØØØ4  3744               3744
ØØØ5  5177               5177              /Ø2:B
ØØØ6  2651               2651
ØØØ7  4136               4136              /Ø3:C
ØØ1Ø  2241               2241
ØØ11  4177               4177              /Ø4:D
ØØ12  3641               3641
ØØ13  5177               5177              /Ø5:E
ØØ14  4151               4151
ØØ15  4477               4477              /Ø6:F
ØØ16  4Ø44               4Ø44
ØØ17  4136               4136              /Ø7:G
ØØ2Ø  2645               2645
ØØ21  1Ø77               1Ø77              /1Ø:H
ØØ22  771Ø               771Ø
ØØ23  7741               7741              /11:I
ØØ24  ØØ41               ØØ41
ØØ25  41Ø2               41Ø2              /12:J
ØØ26  4Ø76               4Ø76
ØØ27  1Ø77               1Ø77              /13:K
ØØ3Ø  4324               4324
ØØ31  Ø177               Ø177              /14:L
ØØ32  Ø3Ø1               Ø3Ø1
ØØ33  3Ø77               3Ø77              /15:M
ØØ34  773Ø               773Ø
ØØ35  3Ø77               3Ø77              /16:N
ØØ36  77Ø6               77Ø6
ØØ37  4136               4136              /17:O
ØØ4Ø  3641               3641
ØØ41  4477               4477              /2Ø:P
ØØ42  3Ø44               3Ø44
ØØ43  4536               4536              /21:Q
ØØ44  3542               3542
ØØ45  4477               4477              /22:R
ØØ46  3146               3146
ØØ47  5121               5121              /23:S
ØØ5Ø  4651               4651
ØØ51  4Ø4Ø               4Ø4Ø              /24:T
ØØ52  4Ø77               4Ø77
ØØ53  Ø177               Ø177              /25:U
ØØ54  77Ø1               77Ø1
ØØ55  Ø176               Ø176              /26:V
ØØ56  74Ø2               74Ø2
ØØ57  Ø677               Ø677              /27:W
ØØ6Ø  77Ø1               77Ø1
                   /
                         EJECT
```

```
   2                          /
Ø1Ø3      ØØ61  1463          1463          /3Ø:X
Ø1Ø4      ØØ62  6314          6314
Ø1Ø5      ØØ63  Ø77Ø          Ø77Ø          /31:Y
Ø1Ø6      ØØ64  7ØØ7          7ØØ7
Ø1Ø7      ØØ65  4543          4543          /32:Z
Ø11Ø      ØØ66  6151          6151
Ø111      ØØ67  77ØØ          77ØØ          /33:[
Ø112      ØØ7Ø  ØØ41          ØØ41
Ø113      ØØ71  1Ø2Ø          1Ø2Ø          /34:\
Ø114      ØØ72  Ø2Ø4          Ø2Ø4
Ø115      ØØ73  41ØØ          41ØØ          /35:]
Ø116      ØØ74  ØØ77          ØØ77
Ø117      ØØ75  4Ø2Ø          4Ø2Ø          /36:CAROT
Ø12Ø      ØØ76  2Ø4Ø          2Ø4Ø
Ø121      ØØ77  Ø1Ø1          Ø1Ø1          /37:UNDERLINE
Ø122      Ø1ØØ  Ø1Ø1          Ø1Ø1
Ø123      Ø1Ø1  ØØØØ          ØØØØ          /4Ø:SPACE
Ø124      Ø1Ø2  ØØØØ          ØØØØ
Ø125      Ø1Ø3  75ØØ          75ØØ          /41:X!
Ø126      Ø1Ø4  ØØØØ          ØØØØ
Ø127      Ø1Ø5  7ØØØ          7ØØØ          /42:"
Ø13Ø      Ø1Ø6  ØØ7Ø          ØØ7Ø
Ø131      Ø1Ø7  2277          2277          /43:POUND SIGN
Ø132      Ø11Ø  7722          7722
Ø133      Ø111  5721          5721          /44:$
Ø134      Ø112  4671          4671
Ø135      Ø113  6462          6462          /45:%
Ø136      Ø114  2313          2313
Ø137      Ø115  5166          5166          /46:&
Ø14Ø      Ø116  Ø526          Ø526
Ø141      Ø117  6ØØØ          6ØØØ          /47:APOSTROPHE
Ø142      Ø12Ø  ØØØØ          ØØØØ
Ø143      Ø121  36ØØ          36ØØ          /5Ø:(
Ø144      Ø122  ØØ41          ØØ41
Ø145      Ø123  41ØØ          41ØØ          /51:)
Ø146      Ø124  ØØ36          ØØ36
Ø147      Ø125  2Ø5Ø          2Ø5Ø          /52:*
Ø15Ø      Ø126  ØØ5Ø          ØØ5Ø
Ø151      Ø127  Ø4ØØ          Ø4ØØ          /53:+
Ø152      Ø13Ø  Ø437          Ø437
Ø153      Ø131  Ø5ØØ          Ø5ØØ          /54:,
Ø154      Ø132  ØØØ6          ØØØ6
Ø155      Ø133  Ø4Ø4          Ø4Ø4          /55:-
Ø156      Ø134  Ø4Ø4          Ø4Ø4
Ø157      Ø135  Ø3ØØ          Ø3ØØ          /56:.
Ø16Ø      Ø136  ØØØ3          ØØØ3
Ø161      Ø137  Ø4Ø2          Ø4Ø2          /57:/
Ø162      Ø14Ø  2Ø1Ø          2Ø1Ø
Ø163      Ø141  4536          4536          /6Ø:Ø
Ø164      Ø142  3651          3651
Ø165      Ø143  21ØØ          21ØØ          /61:1
Ø166      Ø144  Ø177          Ø177
Ø167      Ø145  4523          4523          /62:2
Ø17Ø      Ø146  2151          2151
Ø171      Ø147  4122          4122          /63:3
Ø172      Ø15Ø  2651          2651
Ø173                          /
Ø174                                EJECT
```

```
                  /
Ø151  2414                 2414              /64:4
Ø152  Ø477                 Ø477
Ø153  5172                 5172              /65:5
Ø154  Ø651                 Ø651
Ø155  2516                 2516              /66:6
Ø156  Ø245                 Ø245
Ø157  474Ø                 474Ø              /67:7
Ø16Ø  6Ø5Ø                 6Ø5Ø
Ø161  5126                 5126              /7Ø:8
Ø162  2651                 2651
Ø163  5122                 5122              /71:9
Ø164  3651                 3651
Ø165  22ØØ                 22ØØ              /72::
Ø166  ØØØØ                 ØØØØ
Ø167  46Ø1                 46Ø1              /73:;
Ø17Ø  ØØØØ                 ØØØØ
Ø171  14ØØ                 14ØØ              /74:<
Ø172  4122                 4122
Ø173  1212                 1212              /75:=
Ø174  1212                 1212
Ø175  2241                 2241              /76:>
Ø176  ØØ14                 ØØ14
Ø177  4Ø2Ø                 4Ø2Ø              /77:?
Ø2ØØ  2Ø55                 2Ø55
                  /
                           PMODE
                  /
                           ASMIFZ  DISPLY
                           ASMSKP  476-233
                  /
                  /        CHARACTER DISPLAY
                  /        UNIT CODE = 6Ø FOR ASCII, 61 FOR DIAL CHARACTERS
                  /
22Ø1  4551        IODISP,  JMS I   SETUDF    /USERS FIELD
22Ø2  7321                 ONE STL
22Ø3  Ø4Ø6                 AND I   TMP5      /GET FORMAT BIT
22Ø4  7Ø3Ø        DIM75Ø,  CML RAR           /Ø=ASCII, 1=DIAL
22Ø5  2ØØ6                 ISZ     TMP5      /POINT TO BUFFER CDF
22Ø6  14Ø6                 TAD I   TMP5
22Ø7  3227                 DCA     DICDF     /HOLD FOR BUFFER ADDRESSING
221Ø  2ØØ6                 ISZ     TMP5      /POINT TO BUFFER ADDR
2211  14Ø6                 TAD I   TMP5
2212  3ØØ2                 DCA     TMP1      /HOLD BUFFER ADDR
2213  2ØØ6                 ISZ     TMP5
2214  14Ø6                 TAD I   TMP5      /BUFFER LEN
2215  7Ø4Ø                 CMA               /ALLOW FOR EXTRA DILOOP
2216  743Ø                 SZL               /SKIP IF ASCII
2217  7ØØ4                 RAL               /-2*LEN-1 IF DIAL CODES
222Ø  3ØØ3                 DCA     TMP2
2221  7Ø24                 CML RAL           /GET FORMAT BIT
2222  3ØØ7                 DCA     TMP6      /Ø=DIAL, 1=ASCII
2223  62Ø1                 CDF     Ø
2224  1366                 TAD     DIP4ØØ    /TOP OF SCOPE
2225  3365                 DCA     DIDV
2226  53Ø4                 JMP     DIC43     /CR AND LF
                  /
                           EJECT
```

```
                    /
                    /       MAIN DISPLAY LOOP
                    /
2227  6201  DICDF,  CDF                 /ADDRESS USERS BUFFER
2230  1007          TAD       TMP6      /FORMAT CONTROL
2231  7740  DIM40,  SMA SZA CLA         /FULL ASCII?
2232  5320          JMP       DIASCII   /YES
2233  2007          ISZ       TMP6      /WHICH HALFWORD?
2234  5240          JMP       DILEFT    /LEFT
2235  1402          TAD I     TMP1      /RIGHT
2236  2002          ISZ       TMP1
2237  5246          JMP       DISTRP    /STRIP IT
2240  7240  DILEFT, MONE                /SET SW FOR RGHT
2241  3007          DCA       TMP6
2242  1402          TAD I     TMP1
2243  7012          RTR
2244  /012          RTR
2245  7012          RTR
2246  0154  DISTRP, AND       P77
2247  6201          CDF       0
2250  1372          TAD       DIM47     /COMPARE TO TAB
2251  7450          SNA
2252  5342          JMP       DIC47     /DIAL TAB CODE
2253  1277          TAD       DIP2      /COMPARE TO LF
2254  7450          SNA
2255  5306          JMP       DIC45     /DIAL LINE FEED
2256  1277          TAD       DIP2      /COMPARE TO CR
2257  7450          SNA
2260  5304          JMP       DIC43     /DIAL CAR RET
2261  1370          TAD       DIP3
2262  1231  DIDCHR, TAD       DIM40     /SORT OF RESTORE
2263  0154          AND       P77
2264  7450          SNA                 /IGNORE ZERO
2265  5315          JMP       DILEND
2266  7104          CLL RAL             /TIMES TWO
2267  6141          LINC                /GO TO LMODE FOR DISPLAY
                    LMODE
0270  4002          STC       DIGP      /ADDR OF GRID
0271  2365          ADD       DIDV      /GET VERT
0272  1762          DSC I     DIGP      /DISPLAY THE FIRST GRID
0273  1762          DSC I     DIGP      /AND SECOND
0274  0221          XSK I     HORZ      /INTERCHAR SPACE
0275  0221          XSK I     HORZ
0276  0011  DIP11,  CLR
0277  0002  DIP2,   PDP                 /RETURN TO PMODE FOR LOOP
                    PMODE
2300  1764  DIHCHK, TAD I     DIPH      /CHECK HORIZ
2301  1204          TAD       DIM750
2302  7710          SPA CLA
2303  5315          JMP       DILEND    /OK, GO TO NEXT CHAR
                    /
                    EJECT
```

```
                  /
                  /       CARRIAGE RETURN AND LINE FEED
                  /
2304   1276   DIC43,  TAD     DIP11
2305   3764           DCA I   DIPH     /SET LEFT EDGE
2306   1347   DIC45,  TAD     DIM20
2307   1365           TAD     DIDV     /GO DOWN ONE LINE
2310   3365           DCA     DIDV
2311   1365           TAD     DIDV
2312   1366           TAD     DIP400   /COMPARE TO BOTTOM OF SCOPE
2313   7710           SPA CLA          /OFF SCREEN?
2314   5550           JMP I   MXSKP    /YUP-QUIT NOW
                  /
2315   2003   DILEND, ISZ     TMP2     /CHECK COUNT
2316   5227           JMP     DICDF    /AROUND AGAIN
2317   5550           JMP I   MXSKP    /RETURN
                  /
                  /       GET AN ASCII CHAR
                  /
2320   1402   DIASCII,TAD I   TMP1     /GET NEXT
2321   2002           ISZ     TMP1
2322   6201           CDF     0
2323   1367           TAD     DIM240
2324   7500           SMA              /CONTROL?
2325   5262           JMP     DIDCHR   /NO-DISPLAY IT
2326   1371           TAD     DIBMCR   /CHECK CAR RET
2327   7440           SZA
2330   5334           JMP     .+4
2331   1276           TAD     DIP11
2332   3764           DCA I   DIPH     /SET LEFT EDGE
2333   5315           JMP     DILEND   /GO TO NEXT CHAR
                  /
2334   1370           TAD     DIP3     /CHECK LINE FEED
2335   7450           SNA
2336   5306           JMP     DIC45    /YES
2337   7001           IAC              /HOW BOUT TAB
2340   7640           SZA CLA
2341   5315           JMP     DILEND   /NOT THAT-IGNORE
                  /
                  /       TAB
                  /
2342   1356   DIC47,  TAD     DITABS   /TAB TABLE
2343   3005           DCA     TMP4
2344   1764   DITBLP, TAD I   DIPH     /GET CURRENT CHAR POSITION
2345   7161           CIA STL
2346   1405           TAD I   TMP4     /COMPARE TAB POSITION
2347   7760   DIM20,  SMA SZA SNL CLA /PAST THIS TAB?
2350   5353           JMP     DISTAB   /NO - SET TO THIS
2351   2005           ISZ     TMP4     /POINT TO NEXT TABLE ENTRY
2352   5344           JMP     DITBLP   /LOOP FOR NEXT
                  /
                  /       SET THE TAB POSITION
                  /
2353   1405   DISTAB, TAD I   TMP4     /TAB POSITION
2354   3764           DCA I   DIPH     /BECOMES HORIZ
2355   5300           JMP     DIHCHK   /CHECK EOL
                  /
                  /
                          EJECT
```

```
                      /
                      /       TAB POSITIONS
                      /
2356   2357   DITABS, .+1
                      /
2357   Ø131           131               /TAB
236Ø   Ø251           251               /POSITIONS
2361   Ø371           371               /ON
2362   Ø511           511               /PDP-12
2363   Ø631           631               /SCREEN
                      /
                      /       NEXT LOCATION MUST BE POSITIVE
                      /       AND GREATER THAN 75Ø.
                      /
2364   2ØØ1   DIPH,   HORZ
2365   ØØØØ   DIDV,   Ø
2366   Ø4ØØ   DIP4ØØ, 4ØØ
2367   754Ø   DIM24Ø, -24Ø              /SMA SZA
237Ø   ØØØ3   DIP3,   3
2371   ØØ23   DIBMCR, 4Ø=15
2372   7731   DIM47,  -47
                      /
                      /
                              EJECT
```

```
            /
                    LIST
            /
                    ASMIFZ  RKØ8
                    NOLIST
                    ASMIFZ  RKØ8
                    ASMSKP  762-5Ø4
            /
                    PAGE
            /
            IODISK= .
                    ASMIFZ  RFØ8
                    ASMSKP  7
            /
                    JMS I   SETUDF
                    FOUR            /DISK TYPE?
                    AND I   TMP5
                    CDF     Ø
                    SZA CLA         /SKIP IF RK8
                    JMP I   PDRFØ8  /ELSE RFØ8
            /
            /       RK8 I/O ROUTINE
            /
24ØØ  4545          JMS I   NQUEUE  /QUEUE THIS REQUEST
24Ø1  2413          DKQP
24Ø2  5621          JMP I   DKQED   /IT MUST WAIT
24Ø3  442Ø          JMS I   SETINT  /ENABLE HANDLER
24Ø4  6745          DSKD
24Ø5  2423          DSKINT
24Ø6  4622          JMS I   PSETRW  /SET READ/WRITE BIT
24Ø7  7346          MTHREE
241Ø  3366          DCA     DKRTRY  /SET RETRY COUNT
2411  43Ø1          JMS     DSKDO   /START OP
2412  555Ø          JMP I   MXSKP
            /
            /
2413  ØØØØ  DKQP,   Ø               /CDF TO CURRENT LIST
2414  ØØØØ          Ø               /CURRENT LIST ADDR
2415  ØØØØ          Ø               /BUFFER CDF
2416  ØØØØ          Ø               /BUFFER ADDR
2417  ØØØØ          Ø               /REMAINING COUNT
242Ø  ØØØØ          Ø               /CURRENT BLOCK
            /
            DLDC=6732               /LOAD COMMAND REGISTER
            DLDR=6733               /LOAD DKADDR AND READ
            DLDW=6735               /LOAD DKADDR AND WRITE
            DRDS=6741               /READ STATUS
            DCLS=6742               /CLEAR STATUS
            DSKD=6745               /SKIP ON DONE
            DSKE=6747               /SKIP ON ERROR
            DCLA=6751               /CLEAR ALL
            DLWC=6753               /LOAD WORD COUNT
            DLCA=6755               /LOAD CURRENT ADDRESS
            /
            /
2421  1212  DKQED,  LTQED
2422  1222  PSETRW, SETRW
                    ASMIFN  RFØ8
            PDRFØ8, DRFØ8
            /
                    EJECT
                    /
                    /       DISK INTERRUPT HANDLER
                    /
2423  ØØØØ  DSKINT, Ø
2424  73ØØ          CLA CLL
2425  1213          TAD     DKQP    /USER CDF
2426  3227          DCA     .+1
2427  62Ø1          CDF             /USERS FIELD
243Ø  6747          DSKE            /WAS THERE AN ERROR?
2431  5241          JMP     DKCTU   /NO-CONTINUE
2432  6741          DRDS            /YES-GET STATUS
2433  Ø363          AND     DKFAT   /TEST FATAL BITS
2434  2366          ISZ     DKRTRY  /CHECK RETRY CNT
2435  744Ø          SZA
2436  5351          JMP     DKERR   /FATAL ERROR
2437  43Ø1          JMS     DSKDO   /NO-RETRY
244Ø  5623          JMP I   DSKINT
                    /
2441  2365  DKCTU,  ISZ     DKMORE  /MORE TO GO?
2442  5257          JMP     DKNXT   /NO-GET NEXT FROM Q
2443  1364          TAD     DKLEN
2444  1216          TAD     DKQP+3  /UPDATE ADDRESS
2445  3216          DCA     DKQP+3
2446  1364          TAD     DKLEN
2447  1217          TAD     DKQP+4  /UPDATE LENGTH
245Ø  3217          DCA     DKQP+4
2451  122Ø          TAD     DKQP+5  /OLD STARTING BLOCK
2452  Ø361          AND     M2Ø     /TRUNC TO START OF TRACK
2453  136Ø          TAD     P2Ø     /BEGIN OF NEXT TRACK IS
2454  322Ø          DCA     DKQP+5  /NEXT START BLOCK
2455  43Ø1          JMS     DSKDO   /DO THIS
2456  5623          JMP I   DSKINT
                    /
2457  4546  DKNXT,  JMS I   DQUEUE  /GET NEXT OP
246Ø  2413          DKQP
2461  5271          JMP     DKCLR   /NO MORE-CLEAR
2462  7346          MTHREE
2463  3366          DCA     DKRTRY
2464  1213          TAD     DKQP    /GET NEXT USER FIELD
2465  3266          DCA     .+1
2466  62Ø1          CDF
2467  43Ø1          JMS     DSKDO   /START NEXT
247Ø  5623          JMP I   DSKINT
                    /
2471  6742  DKCLR,  DCLS            /CLEAR STATUS
2472  1156          TAD     M4
2473  1223          TAD     DSKINT  /RETURN-4
2474  3162          DCA     QT1     /POINT TO DSKD IOT
2475  62Ø1          CDF     Ø
2476  116Ø          TAD     KNOP
2477  3562          DCA I   QT1     /CLEAR SETINT
25ØØ  5623          JMP I   DSKINT
                    /
                            EJECT
```

```
            /
            /        THIS DOES THE DISK OPERATION
            /        DATA FIELD IS SET FOR USERS PARAMETERS
            /
2501  ØØØØ  DSKDO,   Ø
2502  6742           DCLS
2503  1614           TAD I   DKQP+1   /READ-WRITE BIT 1
2504  7ØØ6           RTL
2505  72Ø6           CLA RTL          /TO BIT 1Ø
2506  1362           TAD     DKREAD   /MAKE COMMAND
2507  3346           DCA     DKOP
2510  7325           THREE
2511  Ø614           AND I   DKQP+1   /GET UNIT
2512  71Ø4           CLL RAL          /TO BITS 9, 1Ø
2513  1215           TAD     DKQP+2   /BUFFER FIELD
2514  Ø356           AND     P6Ø76
2515  6732           DLDC             /LOAD COMMAND
2516  724Ø           MONE
2517  1216           TAD     DKQP+3   /BUFFER ADDR
252Ø  6755           DLCA             /LOAD CURRENT ADDR
2521  122Ø           TAD     DKQP+5   /STARTING BLOCK
2522  Ø357           AND     P17      /STRIP TO SURFACE-SECTOR
2523  7112           CLL RTR
2524  7Ø12           RTR
2525  7Ø1Ø           RAR
2526  7Ø41           CIA              /MAX POSSIBLE COUNT, THIS TRACK
                                      /LINK CLEAR UNLESS COUNT=4Ø96
2527  3364           DCA     DKLEN    /ASSUMED OPERATION LENGTH
253Ø  724Ø           MONE
2531  3365           DCA     DKMORE   /SET SWITCH
2532  1364           TAD     DKLEN
2533  1217           TAD     DKQP+4   /COMPARE REMAINING REQUEST
2534  762Ø           SNL CLA          /SKIP IF REQUEST LESS OR EQ
2535  5342           JMP     DKWC
2536  1217           TAD     DKQP+4   /REQUEST LEN
2537  7Ø41           CIA
254Ø  3364           DCA     DKLEN    /IS LENGTH OF OP
2541  3365           DCA     DKMORE   /CLEAR SWITCH
2542  1364  DKWC,    TAD     DKLEN
2543  7Ø41           CIA
2544  6753           DLWC             /LOAD WORD COUNT
2545  122Ø           TAD     DKQP+5   /BLOCK NO
2546  6733  DKOP,    DLDR             /READ OR WRITE COMMAND
2547  6747           DSKE             /CHECK IMMEDIATE ERROR
255Ø  57Ø1           JMP I   DSKDO
            /
2551  6741  DKERR,   DRDS             /READ ERROR STATUS
2552  3367           DCA     DKSTAT
2553  6742           DCLS
2554  7327           SIX              /DISK ERROR
2555  5552           JMP I   MERROR
            /
                     EJECT
```

```
Ø746                    /
Ø747     2556   6Ø76    P6Ø76,  6Ø76
Ø75Ø     2557   ØØ17    P17,    17
Ø751     256Ø   ØØ2Ø    P2Ø,    2Ø
Ø752     2561   776Ø    M2Ø,    -2Ø
Ø753     2562   6733    DKREAD, DLDR
Ø754     2563   ØØ36    DKFAT,  36              /FATAL ERR BITS
Ø755     2564   ØØØØ    DKLEN,  Ø
Ø756     2565   ØØØØ    DKMORE, Ø               /-1 IF MORE TO TRANSFER
Ø757     2566   7775    DKRTRY, -3              /RETRY COUNTER
Ø76Ø     2567   ØØØØ    DKSTAT, Ø               /STATUS AT ERROR
Ø761                    /
Ø762                    /
Ø763                    /
Ø764                    /
Ø765                            EJECT
-
```

```
                    /
                            LIST
                    /
                            ASMIFZ  RF08
                            NOLIST
                    /
                            MEND=.!177       /END OF RESIDENT MONITOR
                    /
                            ASMIFN  DIAL
                            NOLIST
                            ASMIFN  DIAL
                            ASMSKP  1522-1220
                    /
                    /       INITIAL LOADING BOOTSTRAP
                    /       ON AIP VOLUMES, THIS IS THE
                    /       FIRST BLOCK OF THE MONITOR.
                    /
                    /       IF LOADED FROM TAPE, IT IS READ INTO 6000-6377,
                    /       AND ENTERED AT 6010.
                    /       IF LOADED FROM DISK, IT MAY BE AT OR BELOW 4001
                    /       IN FIELD 0, AND ENTERED AT RELATIVE 0,
                    /       IT WILL THEN MOVE ITSELF TO 6000.
                    /
                            FIELD   0
                            *6000
                    /
                    /       ENTER HERE IF DISK LOAD
                    /       LOADED ADDRESSES ARE ONE GREATER
                    /       THAN ASSEMBLED ADDRESSES.
                    /
6000    7200                CLA
6001    3343                DCA     IPLDSK+1/SET DISK-LOAD SWITCH
6002    1265                TAD     I16+1   /POINT TO X INFO
6003    3013                DCA     IX3
6004    4353                JMS     IMOVE+1 /COPY THIS TO 6000
                    /
                            *6010
                    /
                    /       ENTER HERE IF TAPE LOAD
                    /
6010    7200                CLA             /CLEAR CHECKSUM FROM AC
6011    1265                TAD     I4015   /POINT TO INFO
6012    3013                DCA     IX3
                    /
                    /       COLLECT NECESSARY INDEX INFORMATION
                    /
6013    7201        INFO,   ONE
6014    1413                TAD I   IX3     /TRUE MON START
6015    3343                DCA     IMST    /SAVE
6016    7240                MONE
6017    1413                TAD I   IX3     /MONITOR LENGTH
6020    7041                CIA
6021    3344                DCA     IMLEN   /MINUS MONITOR LENGTH
6022    7327                SIX
6023    1013                TAD     IX3     /BUMP TO JCL INFORMATION
6024    3013                DCA     IX3
6025    1413                TAD I   IX3     /JCL START BLOCK
6026    3345                DCA     IJST
6027    1413                TAD I   IX3     /JCL LENGTH
6030    3346                DCA     IJLEN
                    /
                            EJECT
```

```
                  /
                  /       NOW WE READ THE MONITOR
                  /
6Ø31   2342               ISZ     IPLDSK   /WAS IT DISK?
6Ø32   527Ø               JMP     ILMDSK   /YES-LOAD MONITOR FROM DISK
6Ø33   1343               TAD     IMST     /START BLOCK
6Ø34   1266               TAD     I4ØØØ    /TRANSFER IN DATA SEG
6Ø35   3241               DCA     ITBLK
6Ø36   6141       ITLOOP, LINC
                          LMODE
ØØ37   Ø64Ø       ITLDF,  LDF     Ø        /SET SEGMENT
ØØ4Ø   Ø7Ø2               RDE              /READ 1 BLOCK
ØØ41   ØØØØ       ITBLK,  Ø                /TAPE BLOCK
ØØ42   ØØØ2               PDP
                          PMODE
6Ø43   7Ø4Ø               CMA
6Ø44   764Ø               SZA CLA          /CHECKSUM CORRECT?
6Ø45   5236               JMP     ITLOOP   /NO
6Ø46   73ØØ               CLA CLL
6Ø47   1241               TAD     ITBLK
6Ø5Ø   1267               TAD     I1ØØ1    /BUMP TBLK&MBLK
6Ø51   3241               DCA     ITBLK
6Ø52   743Ø               SZL              /SAME SEGMENT?
6Ø53   5257               JMP     ITBUMP   /NO=SPECIAL HANDLING
6Ø54   2344       ITLCHK, ISZ     IMLEN    /LAST BLOCK?
6Ø55   5236               JMP     ITLOOP   /NO=GET NEXT
6Ø56   5313               JMP     IBOOT    /YES-SETUP JCBOOT
                  /
                  /
6Ø57   2237       ITBUMP, ISZ     ITLDF    /NEXT SEGMENT
6Ø6Ø   1241               TAD     ITBLK
6Ø61   1266               TAD     I4ØØØ    /SET DF BIT AGAIN
6Ø62   3241               DCA     ITBLK
6Ø63   5254               JMP     ITLCHK
                  /
6Ø64   ØØ16       I16,    16
6Ø65   4Ø15       I4Ø15,  4Ø15
6Ø66   4ØØØ       I4ØØØ,  4ØØØ
6Ø67   1ØØ1       I1ØØ1,  1ØØ1
                  /
                  /       LOAD MONITOR FROM DISK
                  /
6Ø7Ø   6742       ILMDSK, 6742             /DCLS - CLEAR STATUS
6Ø71   135Ø               TAD     IM4ØØ
6Ø72   6753               6753             /DLWC, LOAD WORD COUNT
6Ø73   724Ø               MONE
6Ø74   1351               TAD     IADDR
6Ø75   6755               6755             /DLCA, LOAD CURRENT ADDR
6Ø76   1343               TAD     IMST     /NEXT BLOCK
6Ø77   6733               6733             /DLDR-READ IT
61ØØ   6745               6745             /DSKD-SKIP IF DONE
61Ø1   53ØØ               JMP     .-1
61Ø2   6747               6747             /DSKE-SKIP ON ERROR
61Ø3   53Ø5               JMP     IDNEXT   /NO ERROR-GET NEYT
61Ø4   527Ø               JMP     ILMDSK   /RETRY
                  /
                          EJECT
                  /
                  /       ADDRESS NEXT DISK BLOCK
                  /
61Ø5  1351  IDNEXT, TAD     IADDR
61Ø6  1347          TAD     I4ØØ                /BUMP CORE ADDR
61Ø7  3351          DCA     IADDR
611Ø  2343          ISZ     IMST    /BUMP BLOCK NO
6111  2344          ISZ     IMLEN   /CHECK LENGTH
6112  527Ø          JMP     ILMDSK  /CONTINUE
                  /
                  /
                  /       MONITOR IS IN CORE
                  /
6113  1345  IBOOT,  TAD     IJST    /JCL START BLOCK
6114  3737          DCA I   IJCBP4  /STORE IN CONTROL BLOCK
6115  73Ø5          TWO
6116  1342          TAD     IPLDSK  /BUILD RES UNIT
6117  7ØØ6          RTL
612Ø  7ØØ4          RAL
6121  3736          DCA I   IJCBP   /STORE IN JOBCTL
                  /
                  /
6122  374Ø          DCA I   IPG37P  /CLEAR PAGE 37
6123  234Ø          ISZ     IPG37P
6124  5322          JMP     .-2
                  /
                  /       BEFORE WE GO, CHECK FOR 8K
                  /
6125  6211          CDF     1Ø
6126  7346          MTHREE
6127  3747          DCA I   I4ØØ    /ARBITRARY LOC
613Ø  7325          THREE
6131  1747          TAD I   I4ØØ    /COMPARE
6132  765Ø          SNA CLA         /IS IT THERE?
6133  5425          JMP I   EXIT    /YES, WE CAN GO
                  /
6134  74Ø2          HLT
6135  5334          JMP     .-1
                  /
                  /
6136  Ø553  IJCBP,  JOBCTL          /JOB CONTROL BLOCK POINTER
6137  Ø557  IJCBP4, JOBCTL+4
614Ø  76ØØ  IPG37P, 76ØØ            /PAGE 37 POINTER
6141  1361  IPPQP,  PQP
6142  7777  IPLDSK, -1              /ZERO IF DISK LOAD
6143  ØØØØ  IMST,   Ø
6144  ØØØØ  IMLEN,  Ø
6145  ØØØØ  IJST,   Ø
6146  ØØØØ  IJLEN,  Ø
6147  Ø4ØØ  I4ØØ,   4ØØ
615Ø  74ØØ  IM4ØØ,  -4ØØ
6151  ØØØØ  IADDR,  Ø
                  /
                          EJECT
```

```
            /
            /        MOVE SELF FROM HERE TO 6000
            /
            /        BECAUSE THIS WAS LOADED FROM THE DISK,
            /        THE LOAD ADDRESSES ARE ONE HIGHER THAN
            /        THE CORRESPONDING ASSEMBLED ADDRESSES.
            /
6152  0000  IMOVE,   0
6153  1353           TAD     IMOVE+1 /THIS ADDR
6154  0351           AND     IM400+1 /TRUNCATED TO 400
6155  3010           DCA     IX0     /TO AUTO INDEX
6156  7240           MONE
6157  1372           TAD     I6000+1 /RESULT ADDR
6160  3011           DCA     IX1     /TO AUTO INDEX
6161  1373           TAD     IMVLN+1
6162  3012           DCA     IX2
6163  1410  IMLOOP,  TAD I   IX0     /GET FROM THIS ADDR
6164  3411           DCA I   IX1     /TO 6000 PLUS
6165  2012           ISZ     IX2
6166  5364           JMP     IMLOOP+1/LOOP FOR 400 WORDS
6167  5771           JMP I   .+2     /NOW BEGIN IN EARNEST
6170  6013           INFO
            /
6171  6000  I6000,   6000
6172  7606  IMVLN,   6000-.
            IX0=     10
            IX1=     11
            IX2=     12
            IX3=     13
            /
            /
            /
                     LIST
            /
                     ASMIFN  CREF
                     LISTAPE CREF
            /
            /        MB02
```

```
NO ERRORS
```

```
ABORT   ØØ26
ALLINT  Ø372
BACKSL  175Ø
CARRET  1751
CREF    ØØ14
CR12    ØØØØ
CTLCSW  1771
CZERO   Ø161
DCLA    6751
DCLS    6742
DEQ     11Ø3
DFLNK   Ø316
DIAL    ØØØØ
DIASCI  232Ø
DIBMCR  2371
DICDF   2227
DIC43   23Ø4
DIC45   23Ø6
DIC47   2342
DIDCHR  2262
DIDV    2365
DIGP    2ØØ2
DIHCHK  23ØØ
DILEFT  224Ø
DILEND  2315
DIM2Ø   2347
DIM24Ø  2367
DIM4Ø   2231
DIM47   2372
DIM75Ø  22Ø4
DIPH    2364
DIP11   2276
DIP2    2277
DIP3    237Ø
DIP4ØØ  2366
DISPLY  ØØØ1
DISTAB  2353
DISTRP  2246
DITABS  2356
DITBLP  2344
DKCLR   2471
DKCTU   2441
DKERR   2551
DKFAT   2563
DKLEN   2564
DKMORE  2565
DKNXT   2457
DKOP    2546
DKQED   2421
DKQP    2413
DKREAD  2562
DKRTRY  2566
DKSTAT  2567
DKWC    2542
DLCA    6755
DLDC    6732

DLDR    6733
DLDW    6735
DLWC    6753
DOLLAR  1753
DQCDF   1117
DQUEUE  Ø146
DRDS    6741
DSKD    6745
DSKDO   25Ø1
DSKE    6747
DSKINT  2423
ENQ     1Ø3Ø
ESCAPE  1754
EXIT    ØØ25
FOUR    73Ø7
GRIDS   ØØØØ
HORZ    2ØØ1
IADDR   6151
IBOOT   6113
IDNEXT  61Ø5
IJCBP   6136
IJCBP4  6137
IJLEN   6146
IJST    6145
ILMDSK  6Ø7Ø
IMLEN   6144
IMLOOP  6163
IMOVE   6152
IMST    6143
IMVLN   6172
IM4ØØ   615Ø
INFO    6Ø13
INTADD  Ø467
INTEX   Ø267
INTLEN  Ø676
INTMAX  ØØ1Ø
INTMX4  ØØ4Ø
INTPOP  ØØ3Ø
INTPSH  ØØ27
INTSCN  Ø4ØØ
INTSTK  ØØØ4
INTST4  ØØ2Ø
IOCOM   1Ø1Ø
IODISK  24ØØ
IODISP  22Ø1
IOHLT   1Ø26
IOLTP   12ØØ
IOTFND  Ø672
IOTSCN  Ø653
IOTSLP  Ø66Ø
IOTTY   14ØØ
IPERR1  Ø234
IPERR2  Ø233

IPG37P  614Ø
IPLDSK  6142
IPOPR   Ø236
IPPQP   6141
IPUSHR  Ø212
ISERR   Ø651
ISETR   Ø6ØØ
ISTACK  Ø467
ISTLIM  Ø265
ISTP    ØØ16
ISVAC   Ø333
ISVFLD  Ø346
ISVLNK  Ø345
ISXIT   Ø647
ITBLK   6Ø41
ITBUMP  6Ø57
ITLCHK  6Ø54
ITLDF   6Ø37
ITLOOP  6Ø36
IXØ     ØØ1Ø
IX1     ØØ11
IX2     ØØ12
IX3     ØØ13
I1ØØ1   6Ø67
I16     6Ø64
I4ØØ    6147
I4ØØØ   6Ø66
I4Ø15   6Ø65
I6ØØØ   6171
JCBOOT  1335
JOBCTL  Ø553
JOBENT  34ØØ
KCDF    Ø157
KCIF    Ø352
KLDF    Ø351
KLIF    Ø35Ø
KNOP    Ø16Ø
KP12    ØØØØ
LDLAST  1371
LDLIST  Ø561
LDSTRT  1375
LINFED  1752
LINT    Ø366
LIOF    6ØØ2
LION    6ØØ1
LPØ8    ØØØØ
LRIB    6234
LTBLEN  1334
LTBLK   1272
LTEXIT  1333
LTICLR  1321
LTLINC  Ø45Ø
LTNONE  Ø454
LTOP    1271
LTPCDF  131Ø
LTPDO   1233
LTPINT  1275
```

```
LTPNXT 1315
LTPQP  1214
LTPXOB 1330
LTQED  1212
LTRDE  1332
LT1    1331
LXIF   Ø334
LXJMP  Ø340
MEND   2577
MENTER Ø147
MERR   Ø677
MERROR Ø152
MHALT  ØØ31
MISTAK Ø266
MNOP   ØØ33
MNTR   Ø725
MONE   724Ø
MONSTK ØØØ4
MQUEST Ø713
MSCDEV ØØØØ
MSTACK Ø51Ø
MSTP   ØØ17
MTEMP  ØØØ2
MTHREE 7346
MTWO   7344
MUDF   Ø746
MWAIT  ØØ24
MXDF   Ø772
MXIF   Ø776
MXIT   Ø752
MXSKP  Ø15Ø
MX1    Ø751
M2Ø    2561
M4     Ø156
NQCDF  1Ø47
NQCHN  1Ø72
NQFRST 1Ø65
NQSRCH 1Ø42
NQUEUE Ø145
NQXIT  1Ø55
NTR    Ø2ØØ
ONE    72Ø1
PC12   ØØØØ
PDPIF  Ø31Ø
PECHO  Ø714
PINT   Ø36Ø
PINTAD Ø674
PINTSC Ø675
PMUDF1 1176
PQP    1361
PSETRW 2422
PULIST 1Ø25
PXIF   Ø341
P17    2557
P2Ø    256Ø
P37    Ø347
P5     Ø155

P6Ø76  2556
P7Ø    Ø153
P77    Ø154
P8INT  Ø144
QERR   1152
QMLOOP 1162
QMOVE  1155
QT1    Ø162
QT2    Ø163
QT3    Ø164
QT4    Ø165
RCCP   1362
READ   ØØ21
READR  1ØØØ
RFØ8   ØØØØ
RKØ8   ØØØ1
RWSW   ØØØ7
SETINT ØØ2Ø
SETRW  1222
SETUDF Ø151
SIX    7327
SPCHAR Ø525
TC58   ØØØØ
THREE  7325
TMP1   ØØØ2
TMP2   ØØØ3
TMP3   ØØØ4
TMP4   ØØØ5
TMP5   ØØØ6
TMP6   ØØØ7
TTALT  1741
TTBLEN ØØØ4
TTCLUG 1415
TTCRLF 1731
TTCTLC 1766
TTEBFF 154Ø
TTEBLN 1534
TTEBOP 1532
TTEBUF 1533
TTECHO 15ØØ
TTECHR Ø163
TTECNT 1531
TTECP  1747
TTECTL 1755
TTEOUT 1464
TTEQ   1515
TTETMP Ø162
TTIACT 17Ø3
TTICDF 17Ø5
TTICHR 1632
TTIFSP 17ØØ
TTIINT 1616
TTINRM 164Ø
TTINXT 1662
TTIQ   16ØØ
TTIQER 143Ø
TTIQP  1611

TTISCN 1642
TTISPC 1677
TTITMP 17Ø3
TTIXIT 1672
TTMBLN 7774
TTM14Ø 17Ø7
TTM24Ø 1676
TTM4Ø  165Ø
TTOCDF 1432
TTOCLR 146Ø
TTOFLG 1427
TTOINT 1437
TTONXT 1456
TTOPUT 1431
TTOQ   14Ø2
TTOQP  1421
TTPCTL Ø177
TTRLIM 1746
TTRUB  1712
TTSXIT 1737
TT1ØØ  1765
TT177  1674
TT2ØØ  1675
TWO    73Ø5
ULIST  Ø543
UPARRW 1764
WAIT   ØØ23
WAITR  Ø715
WRITE  ØØ22
WRITR  1ØØ3
XITDF  Ø23Ø
XITIF  Ø231
```