ORNL/TM-5476

Automatic Neutron PSD Transmission from a Process Computer to a Timeshare System

J. B. Bullock
W. H. Sides, Jr.

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

DISCLAIMER

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $5.50; Microfiche $3.00

ORNL/TM-5476
Dist. Category
UC-79 and 79m

Contract No. W-7405-eng-26

INSTRUMENTATION AND CONTROLS DIVISION

AUTOMATIC NEUTRON PSD TRANSMISSION FROM A PROCESS COMPUTER TO A TIMESHARE SYSTEM

J. B. Bullock W. H. Sides, Jr.

Date Published - April 1977

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

PAGES ii to iv

WERE INTENTIONALLY

LEFT BLANK

CONTENTS

1. INTRODUCTION

In other ORNL reports[1,2] the analysis of neutron power spectral density (PSD) data from the High Flux Isotope Reactor (HFIR) is reported. These reports discuss the merits and possible practical applications of PSD analysis and outline the needs for a more extensive data base.

On-line PSD analysis capability was implemented on the original HFIR computer system[3,4] by manually entering several parameters and function request numbers via the operator console. This procedure generated a punched-paper tape which was manually rolled up, labeled, and carried to the PDP-10 facility for entry on an IBM-360 magnetic tape. Because of the inherent errors and inefficiency of manual procedures and a need for larger quantities of data, an automatic process was developed.

2. GENERAL DESCRIPTION OF AUTOMATIC PSD PROCESSING

For economy, software previously developed for PSD processing is used to the maximum extent. Since, originally, the paper-tape data was read into the PDP-10 time-share system at ORNL, and a telephone interface was the routine mode of interacting with the PDP-10, the HFIR computer was equipped with a coupler device to transmit data by telephone. In the overall data acqusition, PSD computation, storage, and transmission system (Fig. 1), an automatic timer--call-in program, FRYLIN, initiates the noise data collection program (MUXBUF) on a fixed time base. The raw data are buffered onto drum storage until the entire data record is complete. Upon completion, MUXBUF schedules BULKRY to calculate the PSD of the data stored in the drum buffer. BULKRY will then write the PSD results and the date, time, and pertinent reactor parameters onto drum storage,

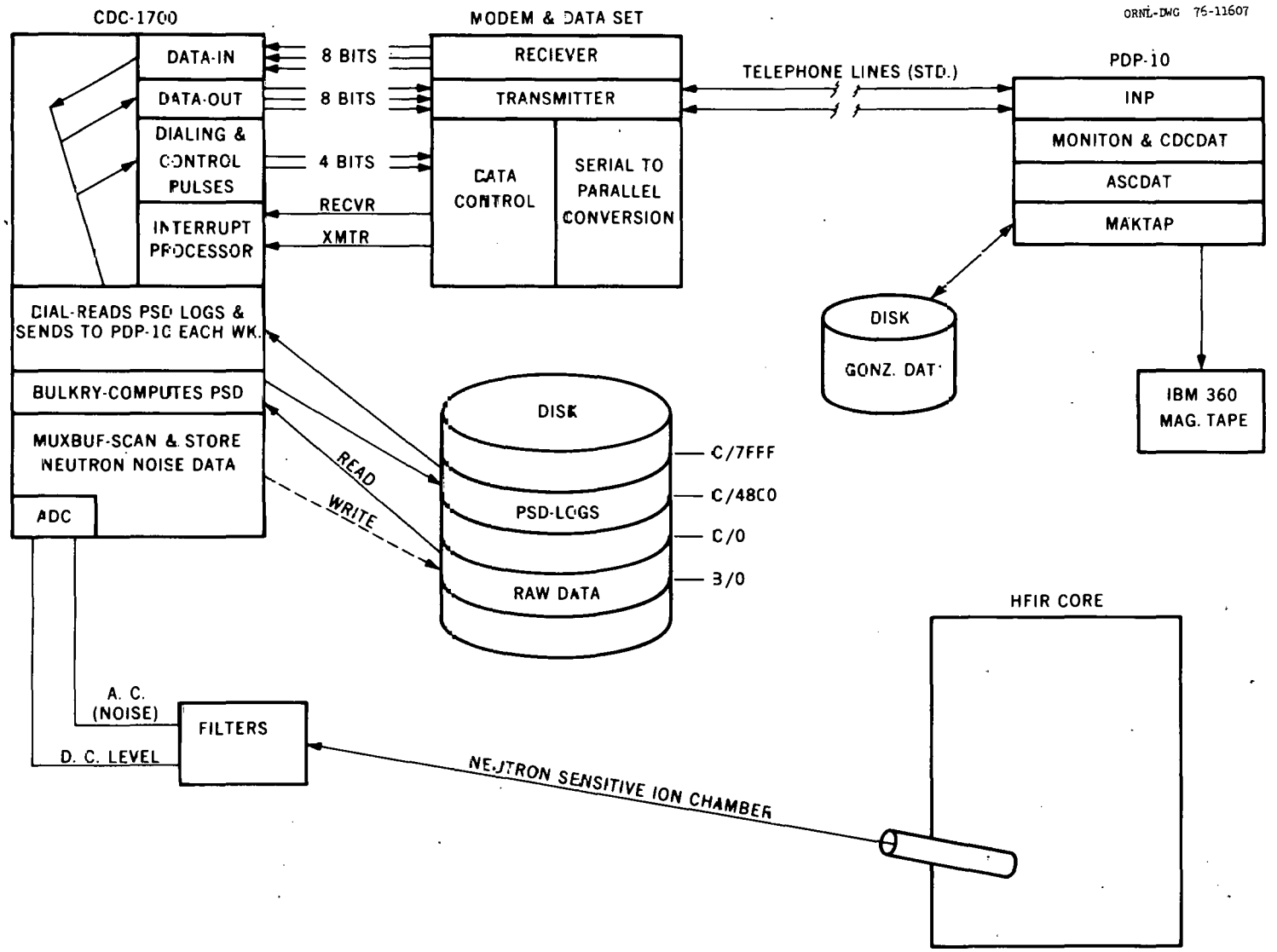


Fig. 1. Overall data collection, computation, and transmission scheme.

starting at address $C/$4800. Each of these records, called a "PSD log," consists of 48 computer words (Table 1). The drum can store 300 PSD logs, which is adequate for a complete 23-day HFIR fuel cycle at a maximum rate of 13 logs per day. After 56 logs have been accumulated, a call is made to program DIAL which calls the PDP-10 by telephone, executes the necessary commands to become a time-share user, transfers all PSD logs from the drum buffer to the PDP-10 disk memory, and properly terminates the time-share dialogue before disconnecting the telephone circuit. The PDP-10 program INP receives the data from the CDC-1700 in 8-bit (half word) bytes and checks for the correct checksum count before storing the logs in an accumulating buffer file labeled GONZ.DAT. Finally, a user initiated program MAKTAP is executed to remove the PSD logs from GONZ.DAT, to convert the binary format into ASCII blocks identical to those previously punched on paper tape by the manual process, and finally to write the ASCII PSD logs onto magnetic tape via an IBM-360. The magnetic tape image generated in this manner can be analyzed by the same pattern recognition algorithms previously developed to run with paper-tape input data.

The following sections describe the programs that implement these functions.

3. PDP-10 PROGRAMS

Two programs were written for the PDP-10 to accomplish its side of the intercomputer data links. Each program comprises one Fortran program and an assembly language (macro) subroutine. One program receives data transmitted by the CDC-1700 through the PDP-10 time share monitor and

Table 1. Power spectral density log format

Word No.	Parameter	Units
1-31	Power spectral density	ABS. Value/const.
32-33	Constant	.XXXE-XX
34	Current year	--
35	Current month	--
36	Current day	--
37	Current hour	--
38	Current minute	--
39	SC reactivity	cents
40	Rod reactivity	cents
41	Spare	--
42	Integrated reactor power	Mw Days × 10
43	Current power level	Mw × 10
44-48	Current rod position	in. × 100

stores it in a compact binary form in a data file, called GONZ, on the PDP-10 user's disk storage area. This program is executed on a daily or weekly basis. The other program retrieves the compact data stored in GONZ and changes it to a form acceptable as input to the IBM 360 computers for creating a magnetic tape. The program is executed by the user when the data storage disk file reaches either near or full capacity or when he considers it appropriate.

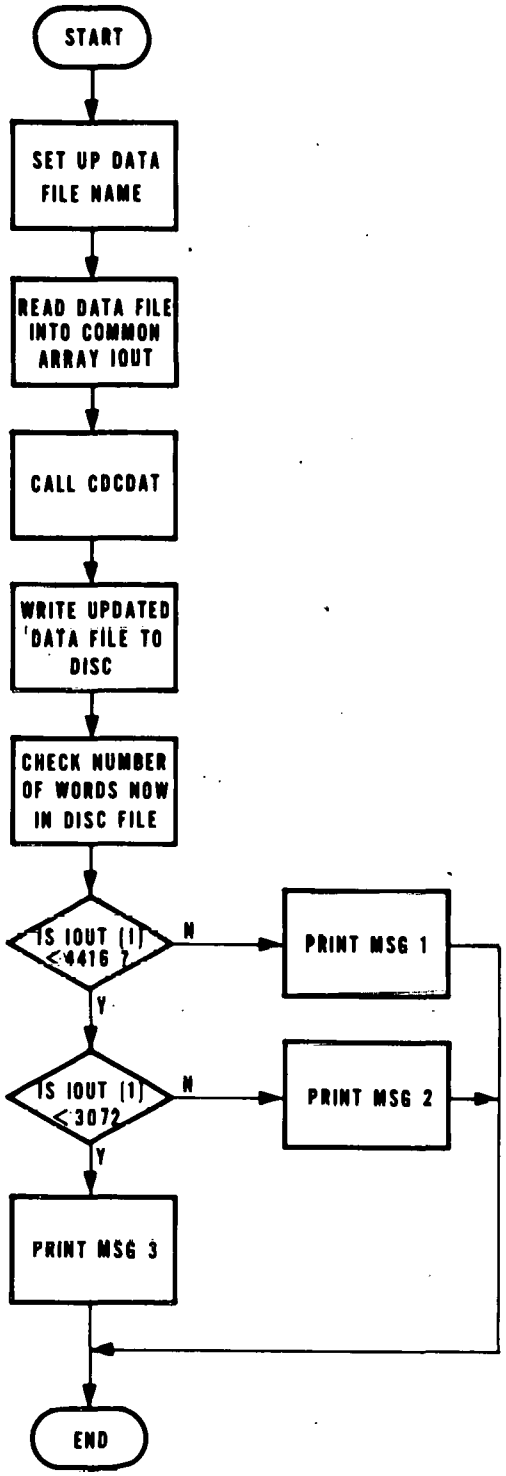
The disk file on the PDP-10 can store 8832 words or data for 184 logs. This is sufficient for logs recorded at 3-hr intervals over the 23-day fuel cycle of the reactor. At this rate, the data retrieval program requires execution only once during the fuel cycle. The magnetic tapes from the IBM-360 computer are combined later to contain data from several fuel cycles. This procedure supplies a large data source for the computer codes used in the reactor diagnostics pattern recognition program at ORNL.

3.1 Data Receiving Program

The Fortran program for receiving and storing data transmitted by the CDC-1700 is INP; the macro subroutine is CDCDAT. After INP (Figs. 2 and 3) reads the fixed-length data storage file GONZ into a 4416-word common array, it then calls subroutine CDCDAT to handle incoming data transmitted by the CDC-1700.

Each transmission consists of one or more logs, a total of forty-eight, 16-bit words of data. Each log sent from the CDC-1700 is a series of ninety-six, 8-bit binary words. On signal from the PDP-10, the CDC-1700 sends to the PDP-10 monitor buffer a group of forty-eight, 8-bit words (one-half of a log). These words are read, one at a time,

ORNL-DWG 76-11609



MSG 1 : MAXIMUM OF 184 LOGS ARE IN GONZ. DAT. DUMP THEM WITH SUBMIT MAKTAP.

MSG 2 : THERE ARE NOW ____ LOGS IN FILE GONZ. DAT. MAXIMUM IS 184 LOGS. START DUMPING SOON WITH MAKTAP.

MSG 3 : THERE ARE NOW ____ LOGS IN GONZ. DAT.

Fig. 2. Logic diagram of INP.

```
      COMMON/ADR/IOUT(4416)
      IDISC=1
      LPT=6
      GONZ=4HGONZ
      CALL IFILE(IDISC,GONZ)
      READ(IDISC,100)(IOUT(I),I=1,4416)
      CALL RELEAS(IDISC)
      CALL CDCDAT
      CALL OFILE(IDISC,GONZ)
      WRITE(IDISC,100)(IOUT(I),I=1,4416)
      END FILE IDISC
      CALL RELEAS(IDISC)
      NUM=IOUT(1)/24
      IF(IOUT(1).LT.4416)GO TO 1
C  IF(IOUT(1).GE.4416)EXECUTE ASCDAT AND THE PL1 PROGRAM WITH MAKTAP
      WRITE(LPT,200)NUM
200   FORMAT(//' DATA FILE GONZ.DAT IS FULL.'/' FILE CONTAINS ',I3,
     1' LOGS.'/' DUMP THEM WITH SUBMIT MAKTAP.'/)
      GO TO 3
1     IF(IOUT(1).LT.3072)GO TO 2
C  IF(IOUT(1).GE.3072)WRITE WARNING MESSAGE
      WRITE(LPT,300)NUM
300   FORMAT(//' THERE ARE NOW ',I3,' LOGS IN FILE GONZ.DAT.'
     1,/' MAXIMUM IS 184 LOGS. START DUMPING SOON WITH MAKTAP.'/)
      GO TO 3
2     WRITE(LPT,400)NUM
400   FORMAT(//' THERE ARE NOW ',I3,' LOGS IN GONZ.DAT.'/)
100   FORMAT(I)
3     END
```

Fig. 3. Source listing of program INP.

from the monitor buffer and assembled in groups of four into twelve, 32-bit binary words. This "stacking" takes maximum advantage of the PDP-10 36-bit word length without further division of the 8-bit groups. A running 16-bit checksum (CHKSUM) is maintained of the incoming 8-bit words. When one-half of a data log has been sent (forty-eight, 8-bit words), a 16-bit checksum is sent from the CDC-1700 in two, 8-bit groups. The checksum is assembled into a 16-bit word and checked against the running 16-bit checksum maintained by the PDP-10 on the incoming data. If the checks agree, a signal for more data is sent to the CDC-1700, and a second series of forty-eight, 8-bit words are transmitted. This is followed by a second, 16-bit checksum. One hundred, 8-bit words are sent from the CDC-1700 to the PDP-10 for each data log: forty-eight, 8-bit words; two, 8-bit checksum words; forty-eight, 8-bit words; and two, checksum words. The 96 data words are assembled into twenty-four, 36-bit words for storage on the PDP-10 disk. The 4416-word data file GONZ allows storage of 184 logs on the disk.

Figure 4 is a diagram of CDCDAT, and Fig. 5 is a source listing. When the subroutine is entered, the location of the first word (named MAX) in the labeled common array ADR contains the pointer which is the current value of the next available location for storing a data word in the 4416-word array. It is also the total number of data words stored in the 4416-word data file, GONZ. Each time data are added to the array, the pointer is updated and maintained in the first word of the file. The location of the first data word in the array is named OUTPUT; the location of OUTPUT indexed by the value of the pointer MAX is the next available storage location for the incoming data.

ORNL DWG. 76-11611

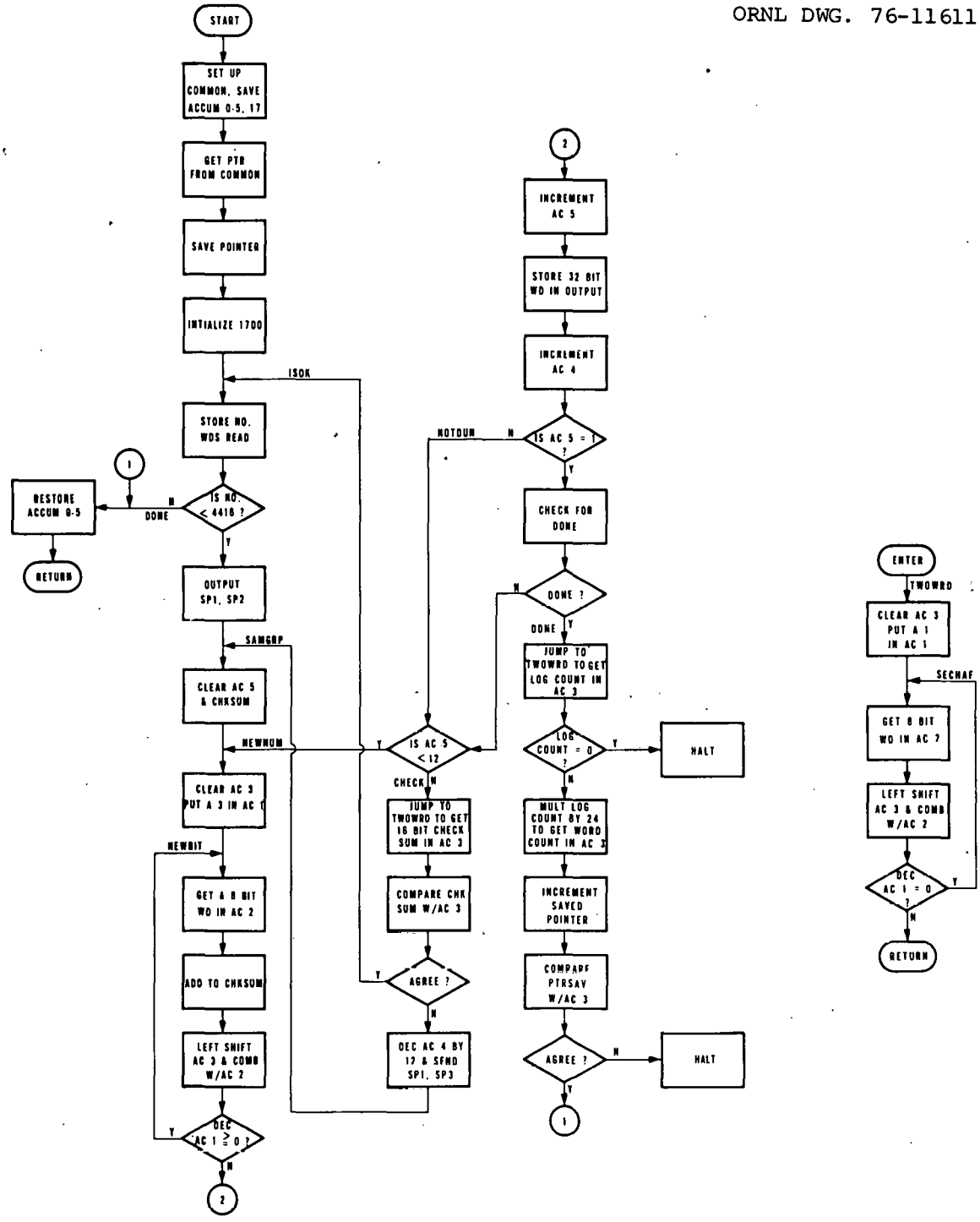


Fig. 4. Logic diagram of CDCDAT.

```
        ENTRY CDCDAT
        EXTERNAL ADR
CDCDAT: 0
        MAX=ADR                         ;POINTER IS FIRST WORD IN COMMON
        OUTPUT=ADR+1                    ;DATA BEGINS IN SECOND WORD
        MOVEM   0,SAV                   ;SAVE AC0
        MOVEM   1,SAV+1                 ;SAVE AC1 & 8 BIT WD INDEX
        MOVEM   2,SAV+2                 ;SAVE AC2 DATA
        MOVEM   3,SAV+3                 ;SAVE AC3 DATA
        MOVEM   4,SAV+4                 ;SAVE AC4 TOT 32 BIT WORD OUTPUT INDEX
        MOVEM   5,SAV+5                 ;SAVE AC5 12 32 BIT WORD INPUT INDEX
        MOVEM   17,SAV+6                ;SAVE AC 17
        MOVE    4,MAX                   ;GET CURRENT POINTER FROM COMMON
        MOVEM   4,PTRSAV                ;SAVE CURRENT POINTER
        JSA     17,INIDEV               ;INITIALIZE 1700 AS A TTY
ISOK:   MOVEM   4,MAX                   ;STORE TOTAL 32 BIT WORDS READ
        CAIL    4,10500                 ;IS MAX GE 4416
        JUMPA   RESTOR                  ;YES, TERM WITH MSG IN FTN PROG
        IONEOU  SP1                     ;OUTPUT SPEC CHAR 1
        IONEOU  SP2                     ;OUTPUT SPEC CHAR 2
SAMGRP: SETZB   5,CHKSUM                ;CLEAR AC5 AND CHKSUM
NEWNUM: SETZ    3,                      ;CLEAR AC3
        MOVEI   1,3                     ;PUT 3 IN AC1
NEWBIT: JSA     17,GETNUM               ;PUT 8 BIT WORD IN AC2
        ADDM    2,CHKSUM                ;FORM 8 BIT CHECKSUM
        LSH     3,10                    ;LEFT SHIFT AC3 8 BITS
        XORM    2,3                     ;COMBINE AC2 WITH AC3
        SOJGE   1,NEWBIT                ;ARE 4 8 BIT WORDS IN
        AOJ     5,                      ;YES, INCREMENT AC5
        MOVEM   3,OUTPUT(4)             ;STORE COMPLETED 32 BIT WORD
        AOJ     4,                      ;INCREMENT AC4
        CAIE    5,1                     ;IS IT TIME TO CHECK FOR DONE
        JUMPA   NOTDUN                  ;NO, GO SEE IF TIME FOR CHECKSUM
        XOR     3,[104236471051         ;CHECK FOR DONE
        JUMPE   3,DONE                  ;JUMP IF DONE
NOTDUN: CAIL    5,14                    ;ARE ALL 12 32 BIT WORDS IN
        JUMPA   CHECK                   ;YES, GO CHECK CHKSUM
        JUMPA   NEWNUM                  ;NO, GET ANOTHER 32 BIT WORD
CHECK:  JSA     17,TWOWRD               ;GET THE 16 BIT CHECKSUM
        XORM    3,CHKSUM                ;CHECK VALUE
        SKIPN   CHKSUM                  ;DO VALUES AGREE
        JUMPA   ISOK                    ;YES, GET NEW GROUP OF 12 32 BIT WDS
        SUBI    4,14                    ;NO, SET OUTPUT INDEX BACK 12
        IONEOU  SP1                     ;OUTPUT SPEC CHAR 1
        IONEOU  SP3                     ;OUTPUT SPEC CHAR 3
        JUMPA   SAMGRP                  ;GET SAME 12 AGAIN
DONE:   JSA     17,TWOWRD               ;GET THE 16 BIT LOG COUNT
        JUMPE   3,NOLOG                 ;HALT IF ZERO LOGS WERE SENT
        IMULI   3,30                    ;CONVERT LOG COUNT TO WORD COUNT
        ADDM    3,PTRSAV                ;ADD WORD COUNT TO OLD VALUE
        AOS     PTRSAV                  ;INCREMENT TOTAL WORD COUNT
        CAME    4,PTRSAV                ;DO WORD COUNTS AGREE
NOLOG:  HALT                            ;NO, HALT
RESTOR: MOVE    0,SAV                   ;RESTORE AC0
        MOVE    1,SAV+1                 ;RESTORE AC1
        MOVE    2,SAV+2                 ;RESTORE AC2
        MOVE    3,SAV+3                 ;RESTORE AC3
        MOVE    4,SAV+4                 ;RESTORE AC4
        MOVE    5,SAV+5                 ;RESTORE AC5
        MOVE    17,SAV+6                ;RESTORE AC17
        JRA     16,(16)                 ;RETURN
SAV:    BLOCK 7
CHKSUM: BLOCK 1
PTRSAV: BLOCK 1
SP1:    52
SP2:    31
SP3:    114
```

Fig. 5. Source listing of subroutine CDCDAT.

```
INIDEV: 0
        OPEN    13,OPNBLK
        HALT
        LOOKUP  13,LKUP
        HALT
        JRA     17,(17)
OPNBLK: 10
        SIXBIT/TTY/
        IBUF
LKUP:   SIXBIT/INPUT/
        SIXBIT/DAT/
        0
        0
IBUF:   BLOCK 3
GETNUM: 0
GET:    SOSGE   IBUF+2
        JRST    GETBF
        ILDB    2,IBUF+1
        JRA     17,(17)
GETBF:  IN      13,0
        JRST    GET
        HALT
TWOWRD: 0
        SETZ    3,              ;CLEAR AC3
        MOVEI   1,1             ;PUT A 1 IN AC1
SECHAF: JSA     17,GETNUM       ;PUT 8 BIT WORD IN AC2
        LSH     3,10            ;LEFT SHIFT AC3 8 BITS
        XORM    2,3             ;COMBINE AC2 WITH AC3
        SOJE    1,SECHAF        ;ARE BOTH 8 BIT WORDS IN
        JRA     17,(17)         ;YES, RETURN
        END
```

Fig. 5. Source listing of subroutine CDCDAT (continued).

Next, the program saves the contents of accumulators 0-5 and 17 for later restoration at exit. The value of the pointer is read from location MAX into accumulator 4 (AC4) and also is saved in location PTRSAV for later error checking before exit. A jump is made to subroutine INIDEV, which declares the CDC-1700 as a Teletype input device for this program. An I/O channel is assigned in the buffered-image mode to allow the PDP-10 to accept the string of 8-bit words exactly as sent from the CDC-1700.

The program then begins a loop. The current value of the pointer is read into location MAX. A check is performed to determine whether the value of the pointer is less than the maximum allowable for the disk file, 10500_8 or 4416_{10} words. If, this maximum is equalled or exceeded, an exit is made to INP. If the current value is less than the maximum, the program sends two special characters, an octal 52 and 31, to the CDC-1700 as a signal to send a group of forty-eight, 8-bit words and their checksum to the PDP-10 monitor buffer. The program then enters a second loop.

The value of the accumulated checksum in location CHKSUM is set to zero, along with AC5. A second loop is entered, and a 3 is placed in AC1, which is used later for an 8-bit word counter. The program then enters the smallest loop, first jumping to subroutine GETNUM to obtain an 8-bit data word in AC2 from the monitor buffer. The subroutine checks its buffer for data, calls for more data from the monitor buffer if necessary, and places the next 8-bit word into AC2. The program then adds the 8-bit word to the running checksum. The contents of AC3 are now left-shifted 8 bits; an exclusive OR is performed with AC2 (the 8-bit data word) which loads the 8-bit word into bits 28-35 (numbering from

zero from the left) in AC3. AC1 (containing a 3) is decremented and checked for greater than or equal to zero.

The program loops back to NEWBIT and jumps to subroutine GETNUM to obtain a second 8-bit data word from the monitor buffer. (The act of obtaining this word clears AC2 of its previous contents.) This word is added to the checksum. AC3 is left-shifted 8 bits, relocating the previous 8-bit word from bits 28-35 to bits 20-27. An exclusive OR with the AC2 puts the new 8-bit word in bits 28-35 of AC3. AC3 now contains both 8-bit data words. AC1 (containing a 2) is again decremented and checked. Thus, this loop is repeated two additional times.

At the termination of the inner loop, the result is four, 8-bit words in AC3 in bits 4-35, and their sum in the checksum. AC5 is incremented to count the number of 32-bit words formed. The contents of AC3 (the assembled 32-bit word) is stored in the common array at the next available location according to the value of the pointer contained in AC4 and the location OUTPUT, as described previously. AC4, the pointer, is incremented.

The contents of AC5, the counter of 32-bit words, is checked to determine whether this is the first (of a group of twelve). If this is the first group, a special check is made of the contents of AC3, the assembled 32-bit word, for the following reason. If the CDC-1700 had no more data to be transmitted, it responded to the two special characters, sent by the PDP-10 requesting data, with DONE in 7-bit ASCII. These four, 8-bit words, assembled into a 32-bit octal word in bits 4-35, appear as octal 10423647105. Therefore, the first 32-bit word in each group of twelve is checked by executing an exclusive OR of AC3 with this

bit pattern. If this pattern is found, the program exits by jumping to the exit routine and returning to the Fortran program. If DONE is not found or if this is not the first 32-bit word in a group of twelve, AC5 is checked to determine whether twelve, 32-bit words had been assembled. If not, the program jumps to NEWNUM and another 32-bit word is assembled in AC3.

When twelve, 32-bit words have been assembled, corresponding to forty-eight, 8-bit words (or one-half a data log), the program jumps to CHECK to verify the checksum. A jump is made to subroutine TWOWRD, which reads and assembles two, 8-bit words from the monitor buffer in the same way in which the 32-bit words are assembled. Upon return, AC3 contains the 16-bit checksum of the first forty-eight, 8-bit words as sent by the CDC-1700. This CDC-1700 version of the checksum is checked against the running sum, accumulated by the PDP-10 in location CHKSUM, by performing an exclusive OR between AC3 and CHKSUM. If the two values do not agree, the pointer in AC4 is decremented by 12 (octal 14), and a second pair of special characters are sent to the CDC-1700. These, an octal 52 and 114, signal the CDC-1700 to repeat the same forty-eight, 8-bit words sent previously. The program then jumps to SAMGRP, which starts through the twelve, 32-bit word gathering loop again. This loop is repeated each time the checksums from the CDC-1700 and the PDP-10 disagree. The CDC-1700 allows several repeats of the data before declaring an uncorrectable error and discontinuing data transmission.

If the checksums agree, the program jumps to ISOK, and the entire process is repeated. Each time twelve, 32-bit words are transferred successfully, the pointer in location MAX is updated.

There were two exit paths from the program. When the bit pattern DONE is encountered in the appropriate place, a jump is made to DONE, the exit routine. This routine receives a 16-bit word from the CDC-1700, containing the number of complete logs transmitted during this execution. The routine first jumps to TWOWRD and obtains the 16-bit log count in AC3 in the same manner in which the checksum value is obtained. A check is made to determine whether this log count is zero. If it is, a halt is executed, which is recognized by the CDC-1700 as an error condition. If a non-zero value is read, the log count is multiplied by 24 (30_8) to convert the log count to a 32-bit word count. This value is added to the value of the pointer, which is saved when subroutine CDCDAT is first entered. This sum, in PTRSAV, is thus the total number of data words residing in the common array, that is, the value of the updated pointer in location MAX. The value of the pointer is also in AC4. However, AC4 had been incremented when the end-of-data signal, DONE, was sent by the CDC-1700. Therefore, the value in location PTRSAV was also incremented by 1 and compared with the value in AC4 (the pointer +1). If the PDP-10 and the CDC-1700 agreed on the number of logs sent, an exit is made by restoring the accumulators to their original saved contents and by returning to the Fortran program. If agreement is not found, a halt is executed to signal an error condition to the CDC-1700.

The end-of-data signal, DONE, is stored as the last data word in the common array. Although the pointer in AC4 is incremented when this word is stored, the value of the pointer in location MAX in the common array remains unchanged. Thus, when the program is executed again, the pointer in location MAX is such that the first 32-bit data word stored overwrites

the DONE from the previous execution. It would be possible, therefore, to locate the last data word in the disk file GONZ by searching for DONE. This would not be necessary, of course, if the value of the pointer in the first word of the file is intact.

The second exit is made when the pointer reaches its maximum allowable value, 4416. In this case, the program jumps to RESTOR, which restores the accumulators to their saved contents and returns to INP.

Upon return, INP writes the updated array back to GONZ on the PDP-10 user disk storage area. The value of the pointer, IOUT(1) in the Fortran common array, is checked and the appropriate message (Fig. 2) is printed on the line printer for delivery to the user. The user receives the messages whenever the automatic execution of the program is initiated by the CDC-1700 and successful transmission of the logs is completed. These messages help inform the user of the amount of data currently in the disk file and indicates to the user when to initiate the retrieval program.

3.2 Data Retrieval Program

The Fortran program for retrieving the packed, binary data from the disk file, GONZ, is ASCDAT, and the macro subroutine is DSKDMP.

ASCDAT is diagrammed in Fig. 6, and Fig. 7 is the source listing. The program reads the data file GONZ from the disk into a 4416-word common array, IOUT. The first word of IOUT is the current number of 32-bit data words in the file. Since each data log consists of twenty-four, 32-bit words, the current number of logs is obtained by dividing by 24. A message is printed on the line printer to record the number of logs in the file. If the number of logs is zero, the program is terminated. If the file contains data, an output disk file, named LOG, is opened.

ORNL DWG. 76-11608

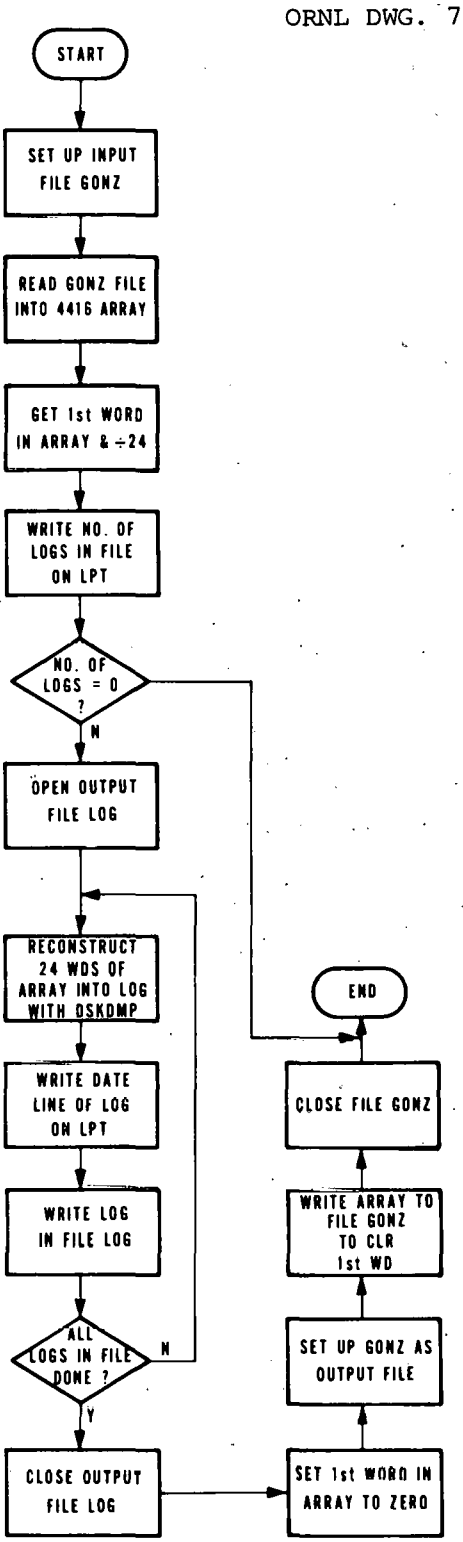


Fig. 6. Logic diagram of ASCDAT.

```
      COMMON/ADR/IOUT(4416),IPSD(31),CONST,IYR,IMO,IDAY,IHR,IMIN,
     1IRHOSC,IRHORD,ISPARE,IACPOW,IAVPOW,IPOS(5)
      DIMENSION FREQ(32),PSD(32)
      IDISC=1
      MAGTAP=1
      LPT=6
      GONZ=4HGONZ
      CALL IFILE(IDISC,GONZ)
C
C  INPUT FILE IS GONZ.DAT
      READ(IDISC,100)(IOUT(I),I=1,4416)
100   FORMAT(I)
      CALL RELEASE(IDISC)
      MAX=IOUT(1)
      MAXNUM=MAX/24
      WRITE(LPT,200)MAXNUM
200   FORMAT(//' THERE WERE ',I3,' LOGS IN FILE GONZ.DAT'//)
      IF(MAXNUM.EQ.0)GO TO 50
      LOG=3HLOG
      CALL OFILE(MAGTAP,LOG)
      DO 10 K=1,MAX,24
      CALL DSKDMP(K)
      WRITE(MAGTAP,12)IMO,IDAY,IYR,IHR,IMIN
      WRITE(LPT,22)IMO,IDAY,IYR,IHR,IMIN
12    FORMAT(6HDATE  ,I2,2X,I2,2X,I2,4X,6HTIME  ,I2,I2)
22    FORMAT(7H DATE  ,I2,2X,I2,2X,I2,4X,6HTIME  ,I2,I2)
      DO 30 I=1,5
      POSIT=FLOAT(IPOS(I))/100.
      WRITE(MAGTAP,13)I,POSIT
13    FORMAT(9HROD POS #,I1,6X,F5.2,2H ")
30    CONTINUE
      AVGPOW=FLOAT(IAVPOW)/10.
      WRITE(MAGTAP,14)AVGPOW
14    FORMAT(10HAVG. POWER,6X,F5.1,3H MW)
      RHOROD=FLOAT(IRHORD)/100.
      WRITE(MAGTAP,15)RHOROD
15    FORMAT(16HROD REACTIVITY $,F5.2)
      RHOSC=FLOAT(IRHOSC)/100.
      WRITE(MAGTAP,16)RHOSC
16    FORMAT(16HSC REACTIVITY  $,F6.2)
      RHODIF=0.
      WRITE(MAGTAP,17)ISPARE
17    FORMAT(16HRHO DIFFERENCE $,I5)
      ACCPOW=FLOAT(IACPOW)/10.
      WRITE(MAGTAP,18)ACCPOW
18    FORMAT(16HACCUM.  POWER    ,F6.1,6H MWDYS)
      WRITE(MAGTAP,19)CONST
19    FORMAT(8HCONST  =,8X,E9.3)
      PSD(1)=0.
      FREQ(1)=0.
      DO 40 I=2,32
      FREQ(I)=FLOAT(I-1)
40    PSD(I)=FLOAT(IPSD(33-I))*CONST
      WRITE(MAGTAP,20)(FREQ(J),J=1,16)
      WRITE(MAGTAP,21)(PSD(J),J=1,16)
      WRITE(MAGTAP,20)(FREQ(J),J=17,32)
      WRITE(MAGTAP,21)(PSD(J),J=17,32)
20    FORMAT(8E9.3)
21    FORMAT(8E9.3)
```

Fig. 7. Source listing for program ASCDAT.

```
10      CØNTINUE
        END FILE MAGTAP
        CALL RELEAS(MAGTAP)
        IØUT(1)=0
        CALL ØFILE(IDISC,GØNZ)
        WRITE(IDISC,100)(IØUT(I),I=1,4416)
        END FILE IDISC
        CALL RELEAS(IDISC)
50      END
```

Fig. 7. Source listing for program ASCDAT (continued).

The program enters a loop which indexes through the 24-word groups in the data file. The loop index is incremented by 24 on each pass. Therefore, the index is also the pointer to the first word of the next 24-word group to be restored as a data log by subroutine DSKOMP. This index is passed to the subroutine as a parameter.

Subroutine DSKDMP is then called. It reads each of the twenty-four, 32-bit words from the disk and places them in proper form in the common array labeled ADR.

The reconstructed data consists of 46 integer numbers and one floating point number for each group of 24 words read. Each integer number is 16 bits in length. The floating point number is 32 bits in length and is located at positions 32 and 33 in each of the 47-word logs. The subroutine places the integer numbers in the proper bit locations for the PDP-10. Since the CDC-1700 is a 16-bit machine and utilizes 1's complement for negative numbers, the subroutine also provides sign extension and conversion to 2's complement for negative numbers for compatibility with the PDP-10.

DSKDMP is diagrammed in Fig. 8, and Fig. 9 is the source listing. The routine first sets up the labeled common ADR in two parts: the first part contains the binary data beginning at location OUT, and the second part is the buffer for the reconstructed data to be output and begins at location DATBUF. The routine then saves the contents of accumulators 1-6 for later restoration at exit. Accumulators 3, 4, and 6 are cleared.

The location of the beginning of the next group of 24 words to be restored as a data log is passed to the subroutine from the Fortran program. This value is placed in accumulator 1. A 2 is placed in accumulator 5 as an index for later use. The first 32-bit word in the 24-word

ORNL DWG. 76-11610

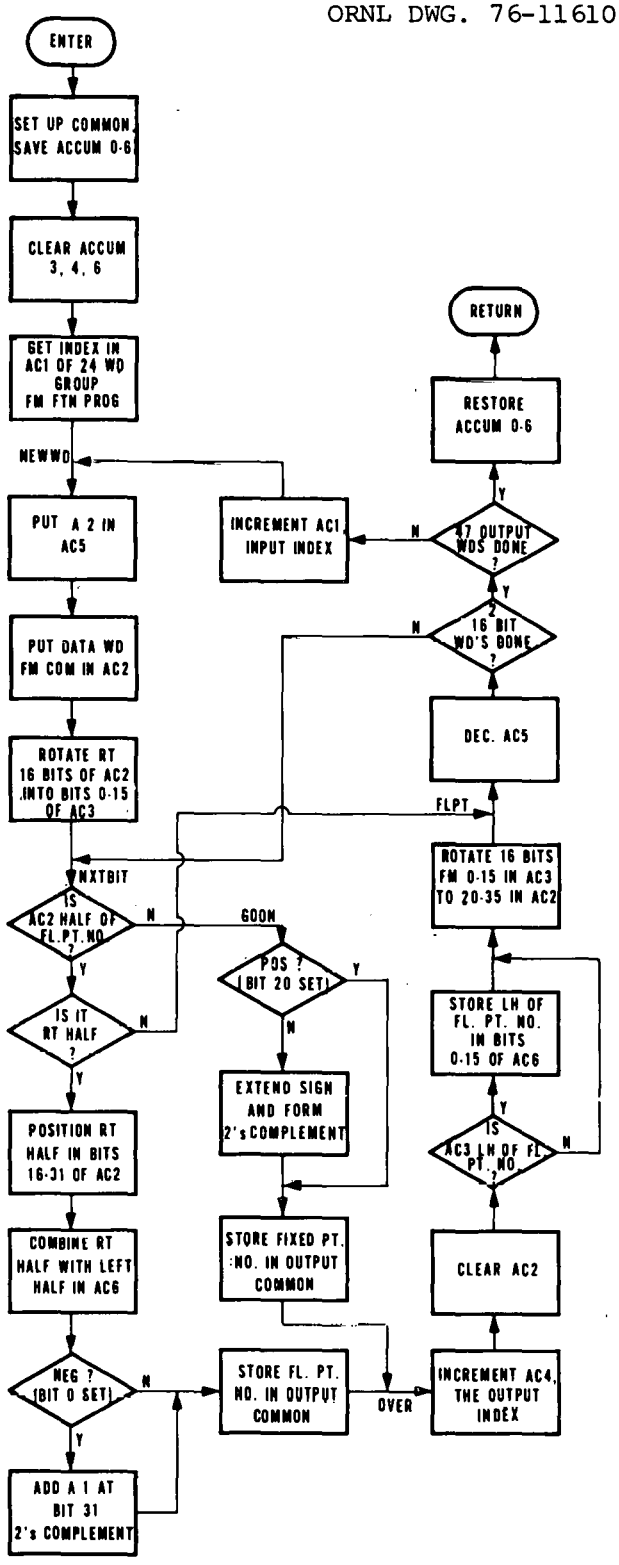


Fig. 8. Logic diagram of DSKDMP.

```
        ENTRY DSKDMP
        EXTERNAL ADR
DSKDMP: 0
        0UT=ADR                 ;SET UP C0MM0N
        DATBUF=ADR+10500        ;10500 0CT = 4416 DEC
        M0VEM   0,SAV           ;SAVE AC0
        M0VEM   1,SAV+1         ;SAVE AC1 INPUT INDEX
        M0VEM   2,SAV+2         ;SAVE AC2 DATA
        M0VEM   3,SAV+3         ;SAVE AC3 DATA
        M0VEM   4,SAV+4         ;SAVE AC4 0UTPUT INDEX
        M0VEM   5,SAV+5         ;SAVE AC5 2 WD INDEX
        M0VEM   6,SAV+6         ;SAVE AC6
        SETZ    6,              ;CLEAR AC6
        SETZB   3,4             ;CLEAR AC3 AND AC4
        M0VE    1,0(16)         ;GET BEGIN 0F 24 WD GR0UP
NEWWD:  M0VEI   5,2             ;PUT 2 IN AC5
        M0VE    2,0UT(1)        ;PUT W0RD IN AC2
        R0TC    2,-20           ;R0TATE 16 BITS INT0 AC3
NXTBIT: CAIE    4,37            ;IS IT DATA 32
        JUMPA   G00N            ;N0, G0 0N
        CAIE    5,2             ;YES, IS IT RIGHT HALF
        JUMPA   FLPT            ;N0, JUMP T0 FLPT
        LSH     2,4             ;YES, P0SITI0N RIGHT HALF
        IOR     2,6             ;C0MBINE WITH LH IN 6
        TLNE    6,400000        ;IS IT NEG
        ADDI    6,20            ;YES, ADD 1 AT BIT 31
        M0VEM   6,DATBUF(4)     ;ST0RE NUMBER IN C0MM0N
        JUMPA   0VER            ;C0NTINUE
G00N:   TRNN    2,100000        ;IS IT P0S
        JUMPA   P0S             ;YES, JUMP T0 P0S
        TD0     2,[777777600000];N0, EXTEND SIGN
        ADDI    2,1             ;TW0'S C0MPLEMENT
P0S:    M0VEM   2,DATBUF(4)     ;ST0RE NUMBER IN C0MM0N
0VER:   A0J     4,              ;INCREMENT AC4
        SETZ    2,              ;CLEAR AC2
        CAIN    4,37            ;IS IT DATA 32
        M0VEM   3,6             ;YES, ST0RE LH IN AC6
        R0TC    2,20            ;N0, R0TATE 16 BITS FR0M AC3 T0 AC2
FLPT:   S0JG    5,NXTBIT        ;ARE B0TH D0NE
        CAIGE   4,56            ;YES, ARE ALL 47 D0NE
        A0JA    1,NEWWD         ;N0, INCR AC1, JUMP
        M0VE    0,SAV           ;YES, REST0RE AC0
        M0VE    1,SAV+1         ;REST0RE AC1
        M0VE    2,SAV+2         ;REST0RE AC2
        M0VE    3,SAV+3         ;REST0RE AC3
        M0VE    4,SAV+4         ;REST0RE AC4
        M0VE    5,SAV+5         ;REST0RE AC5
        M0VE    6,SAV+6         ;REST0RE AC6
        JRA     16,1(16)        ;RETURN
SAV:    BL0CK 7
        END
```

Fig. 9. Source listing of subroutine DSKDMP.

group is loaded into bits 4-35 of accumulator 2. The word is located by indexing from the beginning of the common array by accumulator 1. The 32-bit word in accumulator 2 is right-shifted, placing the rightmost 16 bits in bits 0-15 of accumulator 3 and the leftmost in bits 20-35 of accumulator 2. This divides the 32-bit word into its two 16-bit components. The index of output words in accumulator 4 is checked to determine whether the contents of accumulator 2 are half of the floating point number, that is, output data word 32. (A 31 in accumulator 4 indicates data word 32.) If not, the number in accumulator 2 is an integer number. The sign of the number is checked to determine if it is negative. This is indicated by bit 20, the leftmost of the 16 bits in accumulator 2. If this bit is set, then the original 16 bit number in the CDC-1700 is negative.

Sign extension is then performed by setting bits 0-19. A 1 is added at bit 35 to provide 2's complement. The restored integer number is stored in the common array at location DATBUF, indexed by accumulator 4. Accumulator 4 is incremented. Accumulator 2 is cleared, and a check is made to determine whether the 16 bits previously stored in accumulator 3 are half of the 32-bit floating points number. If not, the 16 bits in bits 0-15 of accumulator 3 are left-shifted into bits 20-35 of accumulator 2. Accumulator 5 is decremented by 1 and checked for zero. If the decrementing accumulator 5 does not produce a zero, then both 16-bit integer numbers contained in the 32-bit word have not been processed and the routine jumps to NXTBIT to restore the second integer number. If both had been done, the routine checks the index in accumulator 4 to determine whether all 47 output numbers have been

processed. If they have, accumulators 0-6 are restored to their original contents, and an exit is made to the Fortran program. If they have not, the routine increments accumulator 1 (input word index) and jumps to NEWWD to obtain the next 32-bit word from the common array.

When the floating point number is encountered in the data, the first half is located in the second half of the sixteenth word in the 24-word group. The second half is located in the first half of the seventeenth word. After the first 16 bits of a data word are processed and stored in the output array, a check is made to determine whether the other 16 bits in accumulator 3 are the first (left) half of the floating point number. If so, this number is saved in bits 0-15 of accumulator 6 for later use, and the routine is jumped to NEWWD for a new 32-bit word in accumulator 2. The first half of this word is the second (right) half of the floating point word. After right-shifting the rightmost 16 bits of accumulator 2 into bits 0-15 of accumulator 3, the 16 bits remaining in accumulator 2 are positioned into bits 16-31. This is the proper location of the right half of the floating point number. An exclusive OR is performed to combine the right half in bits 16-31 in accumulator 2 with the left half in bits 0-15 previously stored in accumulator 6. A check is made of bit 0 in the combined floating point number in accumulator 6 to determine whether the number is negative. If so, a 1 is added at bit 31 to form the 2's complement for compatibility with the PDP-10. The completed floating point number is stored in CONST of the output array.

After restoration of the remaining integer numbers, the routine restores the accumulators to their previously saved contents and exits

to the Fortran program. The date line from the completed log is printed on the line printer for later identification by the user. The entire log with the appropriate alphanumeric labels is output to the disk in the output file LOG. A sample log as it appears on the line printer is shown in Fig. 10.

After all logs in the disk file GONZ are restored in disk file LOG, the first word in the array is set to zero. The entire array is read back to the disk as file GONZ. This zeroes the pointer in file GONZ without destroying the data. Thus, as the CDC-1700 sends more data to be stored on the PDP-10 disk in file GONZ, the old data are overwritten. The program closes the disk file and terminates.

The data retrieval from the binary disk file is initiated by executing the sequence of commands to the PDP-10 contained in file MAKTAP.CTL (Fig. 11). The command sequence is initiated by typing SUBMIT MAKTAP after logging in on the PDP-10 time-share system. The first command in the sequence is execution of the ASCDAT program. This program creates the file LOG.DAT, containing all of the logs reconstructed from the binary data saved on the PDP-10 disk.

The second command causes the current amount of available disk storage to be printed on the line printer for the user. The third command queries the rest of the job to the IBM-360 computer. When executed on the IBM-360, the first program, RCG.PL1 (Fig. 12), reads the file LOG.DAT from the PDP-10 disk to the IBM-360 in 80-column card image form. This program is written in PL1 language. The next program, RCG.SRT (Fig. 13), sorts the logs into chronological order, using the date and time each log was taken.

```
DATE   5   9  74     TIME  13 7
RØD PØS #1        18.11 "
RØD PØS #2        18.13 "
RØD PØS #3        18.13 "
RØD PØS #4        18.07 "
RØD PØS #5        18.13 "
AVG. PØWER         61.8 MW
RØD REACTIVITY $13.93
SC REACTIVITY  $ -0.05
RHØ DIFFERENCE $    0
ACCUM.  PØWER       0.4 MWDYS
CØNST  =          .119E-09
.000E+00 .100E+01 .200E+01 .300E+01 .400E+01 .500E+01 .600E+01 .700E+01
.800E+01 .900E+01 .100E+02 .110E+02 .120E+02 .130E+02 .140E+02 .150E+02
.000E+00 .567E-06 .338E-06 .443E-06 .114E-05 .203E-05 .237E-05 .186E-05
.233E-05 .991E-06 .692E-06 .530E-06 .437E-06 .224E-06 .361E-06 .428E-06
.160E+02 .170E+02 .180E+02 .190E+02 .200E+02 .210E+02 .220E+02 .230E+02
.240E+02 .250E+02 .260E+02 .270E+02 .280E+02 .290E+02 .300E+02 .310E+02
.285E-06 .140E-06 .116E-06 .868E-07 .992E-07 .188E-06 .116E-06 .501E-07
.912E-07 .345E-07 .330E-07 .299E-07 .407E-07 .308E-07 .342E-07 .409E-07
```

Fig. 10. Sample log.

```
.RUN ASCDAT
.R GU0LST
.R SUBMIT
*//GØNZE001 JØB (11114),'RC GØNZALES 3500',CLASS=E
*=RCG.PL1
*=LØG.DAT
*/*
*=RCG.SRT
*=RCG.PRT
*//
*ENDINPUT
.RENAME LØG.TMP<000>=LØG.DAT
.R PIP
*[2,20]/X=*MAKTAP.MSG
```

Fig. 11. Command sequence MAKTAP.CTL.

```
/*ROUTE  PRINT LOCAL
//   EXEC PL1LFCLG,PARM.PL1L='STMT'
//PL1L.SYSPRINT DD SYSOUT=A
//PL1L.SYSIN DD *
 EDIT:    PROC OPTIONS(MAIN) ;
          DCL CARD CHAR(80) VAR, CASE CHAR(1000) VAR,
              OUT CHAR(1000), MORE BIT(1) INITIAL('1'B),
              C(80) CHAR(1) BASED(P) ;
          ON ENDFILE(SYSIN) BEGIN ;
                            MORE='0'B ;
                            GO TO OUTPUT ;
                            END ;
          P=ADDR(CARD) ;
          GET EDIT(CARD) (A(80)) ;
          DO I=80 TO 1 BY -1 WHILE(C(I)=' ') ;
          END ;
          CASE=SUBSTR(CARD,1,I+1) ;
 INPUT:   GET EDIT(CARD) (A(80)) ;
          DO I=80 TO 1 BY -1 WHILE(C(I)=' ') ;
          END ;
          IF I=0 THEN GO TO INPUT ;
          IF I<80 THEN I=I+1 ;
          CARD=SUBSTR(CARD,1,I) ;
          ID=INDEX(CARD,'DATE') ;
          IF ID=0 THEN DO ;
                       CASE=CASE||CARD ;
                       GO TO INPUT ;
                       END ;
 OUTPUT:  PUT SKIP EDIT(CASE) (A) ;
          OUT=CASE ;
          WRITE FILE(TAPE) FROM(OUT) ;
          CASE=SUBSTR(CARD,ID) ;
          IF MORE THEN GO TO INPUT ;
      END EDIT ;
/*
//LKED.SYSIN DD *
/*
//GO.TAPE DD DSNAME=&&REACTOR,UNIT=SYSDA,DISP=(NEW,PASS),
//   DCB=(RECFM=F,BLKSIZE=1000),SPACE=(1000,(500,50))
//GO.SYSIN DD *
```

Fig. 12. Source listing of RCG.PL1.

```
//ØRDER EXEC SØRT
//SØRT.SØRTIN DD DSNAME=&&REACTØR,DISP=(ØLD,DELETE)
//SØRT.SØRTØUT DD UNIT=TAPE9,DISP=(ØLD,PASS),VØL=SER=58,
//  LABEL=(,NL),
//  DCB=(RECFM=F,BLKSIZE=1000)
//SØRTWKØ1 DD UNIT=SYSDA,SPACE=(CYL,(10),,CØNTIG)
//SØRTWKØ2 DD UNIT=SYSDA,SPACE=(CYL,(10),,CØNTIG)
//SØRTWKØ3 DD UNIT=SYSDA,SPACE=(CYL,(10),,CØNTIG)
//SØRT.SYSIN DD *
 SØRT FIELDS=(15,2,A,7,2,A,11,2,A,27,4,A),FØRMAT=CH
 RECØRD TYPE=F,LENGTH=(1000)
/*
```

Fig. 13. Listing for RCG.SRT.

The next program, RCG.PRT (Fig. 14), prints the sorted logs on the line printer for the user. The next command renames the LOG.DAT file to LOG.TMP so that it will be deleted by the PDP-10 monitor after use by the RCG programs. The final command types the contents of file MAKTAP.MSG on the computer operator console teletypewriter. The contents of this file is the job submission form shown in Fig. 15.

4. CDC-1700 PROGRAM DESCRIPTIONS

Additional programs required for the CDC-1700 computer are FRYLIN and DIAL. The complete assembly language listing for each program is shown in the Appendix, Sect. 5.1. The logic block diagram for each program is shown in Figs. 16 through 23. The summary of significant features presented below is a description of the logic diagrams. A summary of the programmed response to the normal PDP-10 messages is shown in the Appendix, Sect. 5.2. The possible error messages are listed in the Appendix, Sect. 5.3.

4.1 Summary of Significant Features of FRYLIN

1. Initially scheduled by GIDUP[3] and recalled by self-contained timer call every 1.5 hr (Fig. 16).

2. Type error message -U; call back in 1.5 hr whenever the least significant bit (LSB) of drum address for data buffer is >$6800 or <$4800. The current LSB is stored in common location $7F41.

3. When the process scan is *off*, that is, location $7F00 = 0, call DIAL to transfer all saved logs to PDP-10. When the scan is *on*, a transfer is attempted only if more than 56 logs are accumulated and the time is between midnight and 6 AM.

```
//  EXEC UCHBUF
//G0.FT5OF001 DD *
BCD
DUMP
/*
```

Fig. 14. Listing for RCG.PRT.

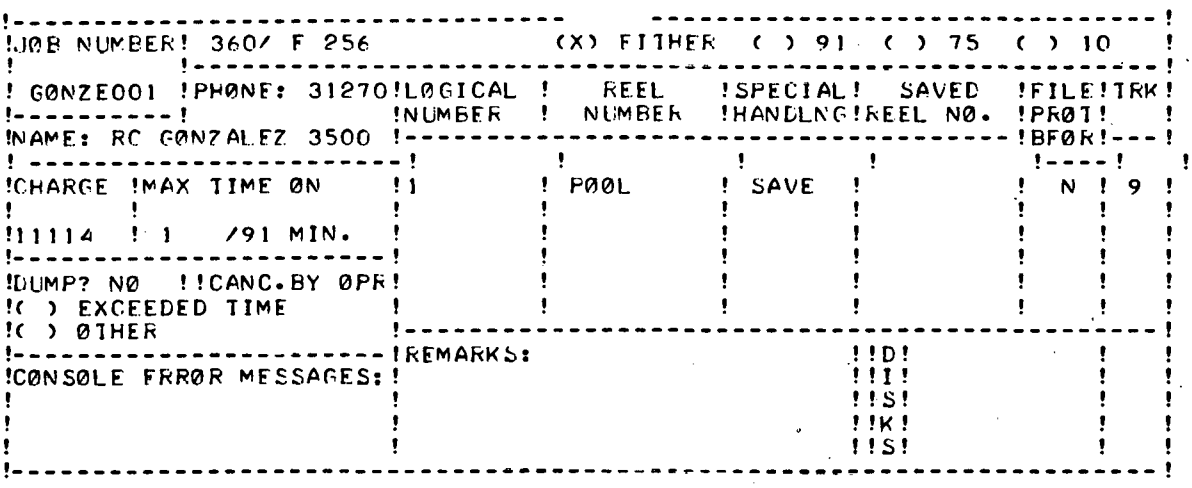


Fig. 15. Job submission form in file MAKTAP.MSG.

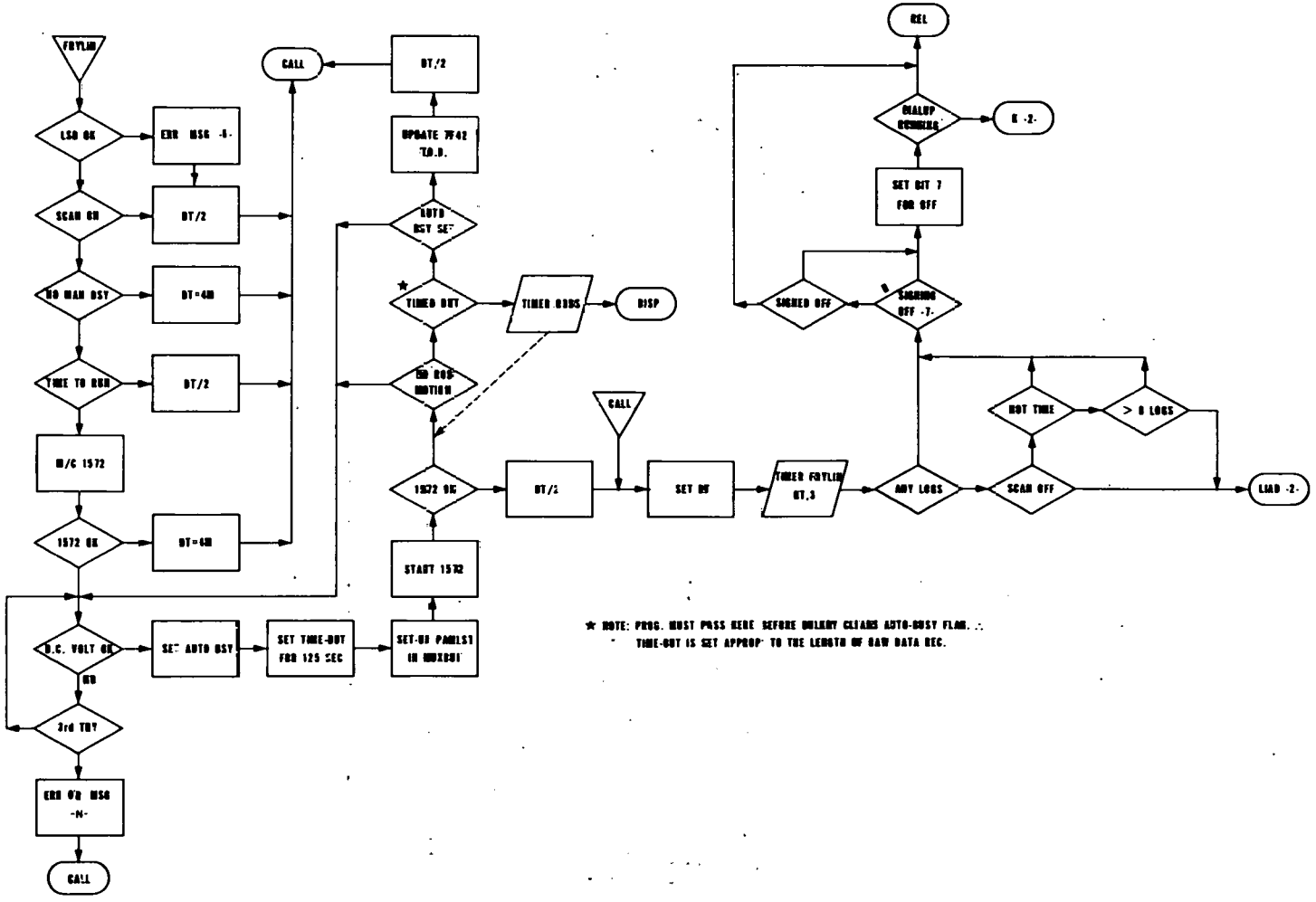


Fig. 16. Program FRYLIN.

4. Determine that no manual use of the PSD programs is currently in progress. AUTBSY and MANBSY are flags in MUXBUF to indicate when automatic or manual programs are in service. Each flag is cleared by BULKRY execution.

5. The 1572 sample rate generator is set to scan at 128 samples/sec. The parameters are set in the MUXBUF table, PAMLST, to save 12,800 points, starting at drum $B/0000.

6. Before starting the 1572 sample rate generator, a test of the input signal dc level will result in error message -N when out of bounds.

7. During the operating period of the 1572 sample rate generator, a check is made every 16 msec for manual actuation of the reactor control rod "raise-lower" switch. If detected, the data scan is restarted.

8. A memory location "MORE" is incremented each 16.66 msec and tested for overflow. By presetting MORE = $62B3, the overflow condition will occur 125 sec after the 1572 is started. This will occur after all data have been digitized and before BULKRY has finished and thus nulled AUTBSY. Therefore if AUTBSY is zero, it is assumed that MUXBUF has halted because the input signal is out of range. This will cause a new scan sequence to be initiated, as in the case above.

9. When none of the above errors causes a restart, the program will update $7F42 with the current hour value, which is used in subsequent executions to determine that 3 hr has elapsed since the last log was saved.

10. The status word (Table 2) at $7F4E is checked and if bit-7 (signing-off) is found set or bit-6 (signed-off) is not set a call for DIAL will be made if one is not currently in the core and the job number status checks below are satisfactory.

11. To aid in autorecovery after system crashes or telephone disconnects, a series of checks is made on the status words in low-core location $6E and common location $7F4E. Normally the two words will be identical; however, a restart of the CDC-1700 will cause CORSTA ($6E) to be nulled. The common location $7F4F is used to save the current job number assigned by the PDP-10 monitor when the computers are satisfactorily linked. This information is used in the LIAD section of FRYLIN to determine which mode of connection to the PDP-10 should be attempted (Fig. 17).

    a. If a DIAL is currently in the core (second word of system library $\neq 0$), FRYLIN releases and waits for the next timer call.

    b. When core status = common status, that is, CORSTA = COM, and is not logged in (bit 15 = 0) and Job No. = 0, then clear the attaching request (bit 10), set the logging-in request (bit 5), and call DIAL. This will generate a normal request to the PDP-10, that is, LOGIN (6415, 26).

    c. If a nonzero job number had been detected in the preceding step, it would be tested for range ($0 < \# < 60$) and ATTACH (bit 10) would be requested for this job number. A bad number will cause "error message-A" and an attempt to get STATUS[6415,26].

Table 2. Status word bit assignment contained in $7F4E

Bit No.	Condition indicated
15	Logged-in as PDP-10 user
14	Spare
13	Running INP
12	Outputting PSD logs
11	Spare
10	Attaching
9	Spare
8	Spare
7	Signing off
6	Signed off
5	Logging-in requested
4	Spare
3	Dialing or connecting in progress
2	Sending password
1	Checking status
0	STRIN (data or message in progress)

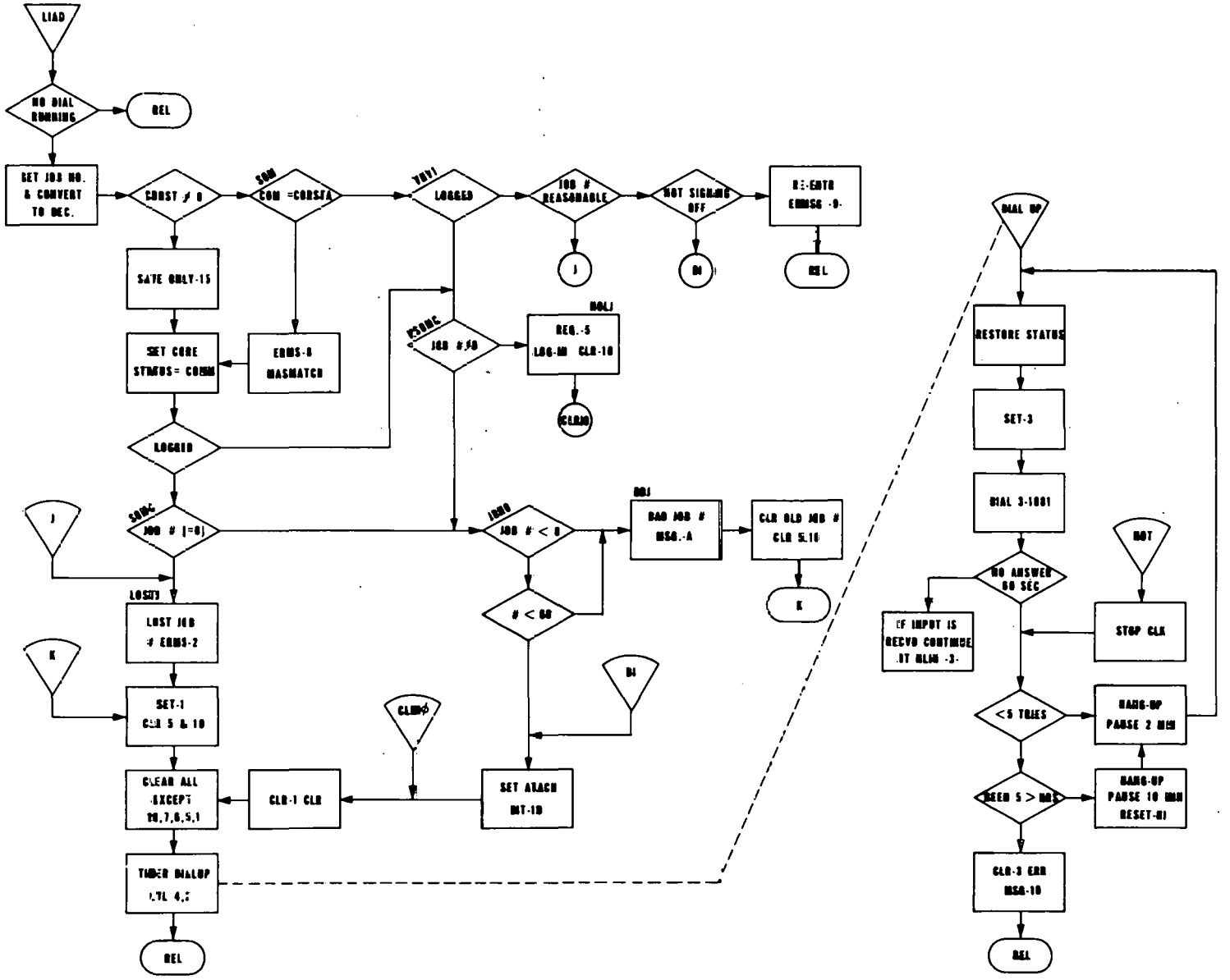


Fig. 17. LIAD portion of FRYLIN.

d. If the initial status indicates "logged-on" but the job number is out of range, a status will also be requested after typing "error message-2."

e. When CORSTA is found null, a CDC-1700 crash is assumed. Only bit 15 of COMSTAT is retained, and the two status words are equated. If the logged-in bit is set, an ATTACH will be requested unless the job number is bad, in which case STATUS is requested.

4.2 Summary of Significant Features of DIAL

1. DIAL is called by FRYLIN whenever a telephone connection to the PDP-10 is required (Fig. 17).

2. The status word on initial entry is saved and reused if some condition during the execution of DIAL results in a disconnect and a need to redial.

3. If the PDP-10 does not answer the dial within 60 sec, the number is redialed after a 2-min pause. This sequence repeats five times before pausing 10 min. If no answer is received within 5 hr, the program types "error message-1" and releases.

4. DIAL is designed to use the normal PDP-10 time-share monitor command string for logging, attaching, etc. This is accomplished by saving all the characters sent by the PDP-10 and responding with the appropriate ASCII character string.

5. When the data set has received a character from the PDP-10, an interrupt on-line 14 starts execution at FETCH (Fig. 18). If the last message sent to the PDP-10 was not RUINP, then an ASCII conversation mode is in use. In this mode the parity bit is dropped,

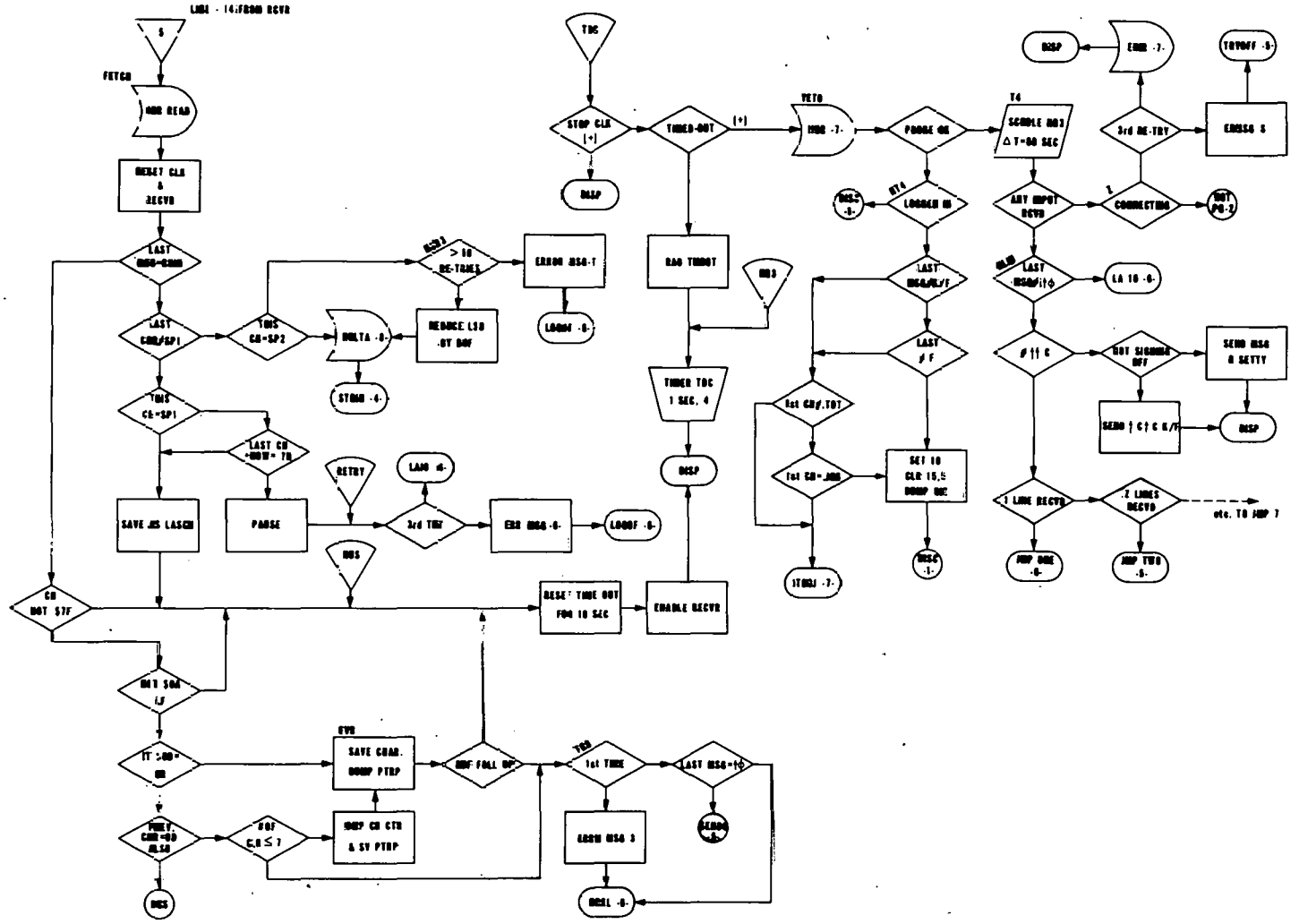


Fig. 18. FETCH portion of DIAL.

line feed and null are ignored, multiple carriage returns are counted as one, all characters preceding the first detected carriage return (CR) are ignored, and all characters between the first CR and last character in a logical record are stored for later interpretation. The logical record is determined by a 10-sec lapse after the last character is received.

6. When the last character (LASCH) is detected, a branch is calculated to the proper subroutine, based on the number of lines in the message received from the PDP-10 (Figs. 19-23). There are two exceptions to this rule. (1) The first exception is that after sending the password during a normal log-in sequence, a long message normally will be received unless a ↑ϕ is sent and the status bit 15 is set while bits 2 and 6 are cleared. If any logical record is received from the PDP-10 in response to ↑ϕ, a jump to LA10 will either start the data transfer (RUINP) or sign-off (↑C↑CK/F) depending on whether bit 7 (signing-off) is clear or set. (2) The second exception is the response to ↑C↑C. This message is sent to the PDP-10 after any attaching sequence. Since attaching will be made to any TTY unit assigned by the PDP-10 monitor, the mode may not be correct for DIAL; therefore, R SETTTY will be sent if the signing-off bit is not set.

7. Prior to an interpretation of any response from the PDP-10, the telephone link is checked. If a dial-tone is detected, a disconnect is assumed to have occurred. If STATUS shows "not logged in," the disconnect procedure is initiated, as in the no-answer case above. If logged in and not signing-off with K/F or F, then the attach bit

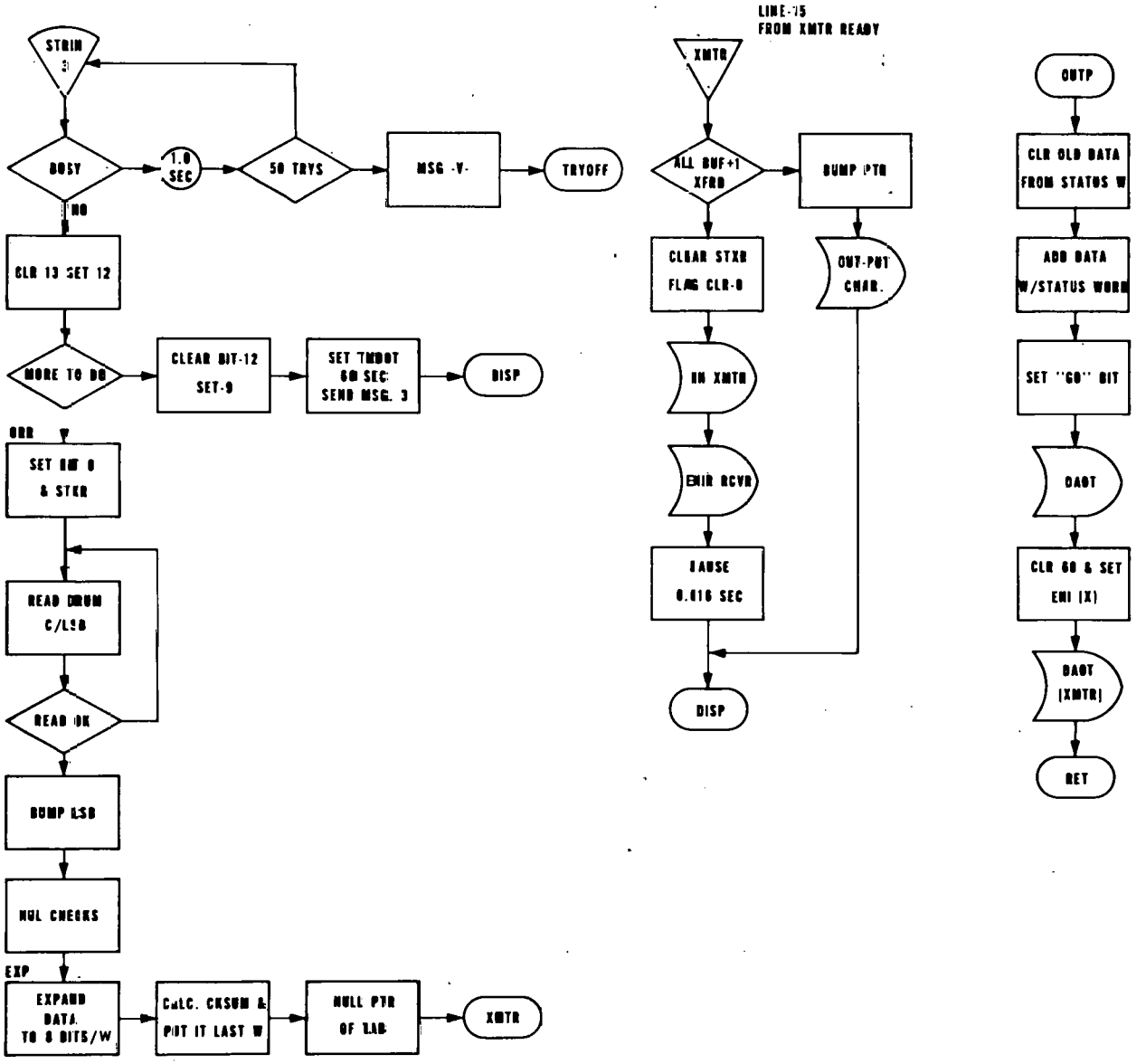


Fig. 19. Subprograms STRIN, XMTR, and OUTP.

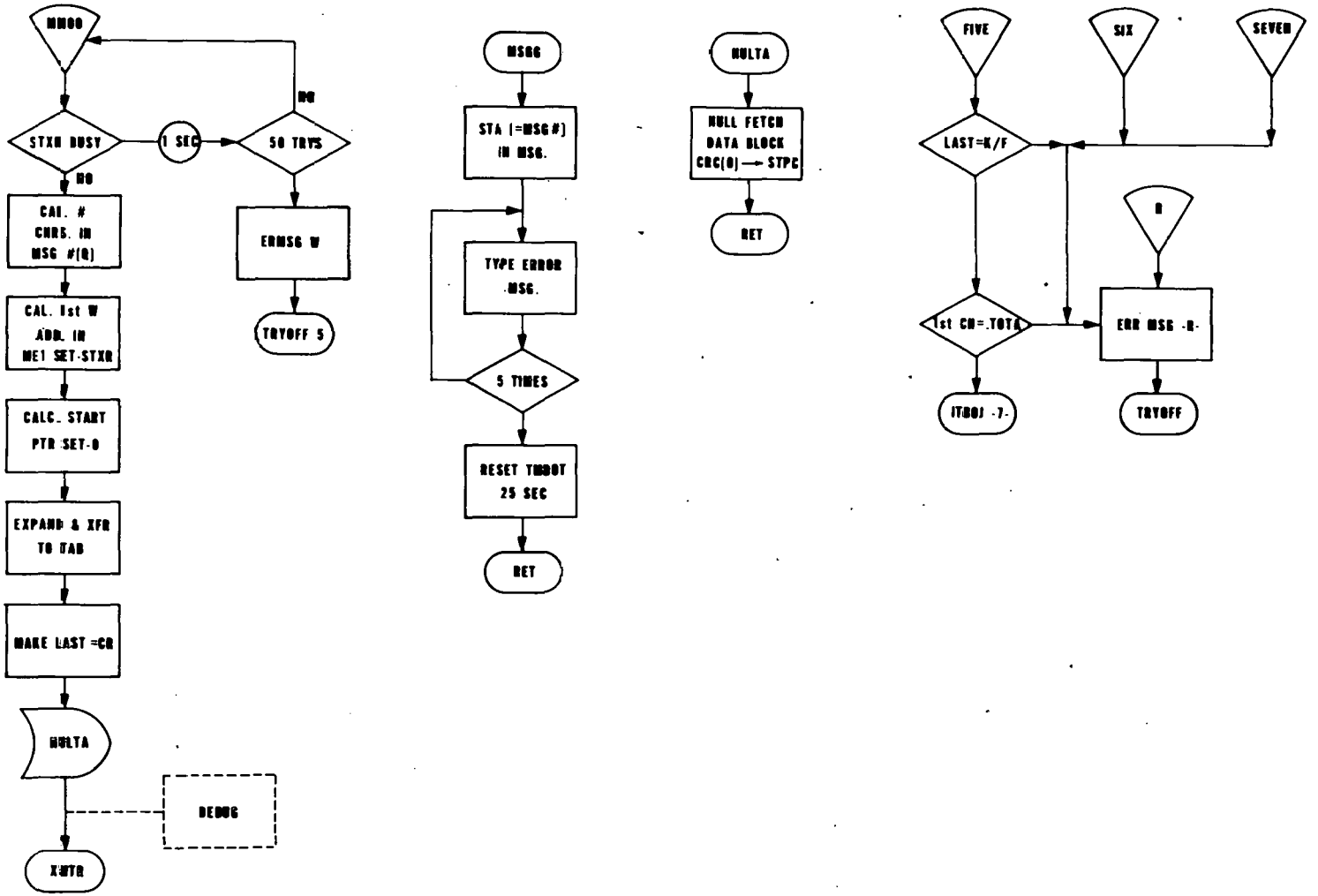


Fig. 20. DIAL subroutines.

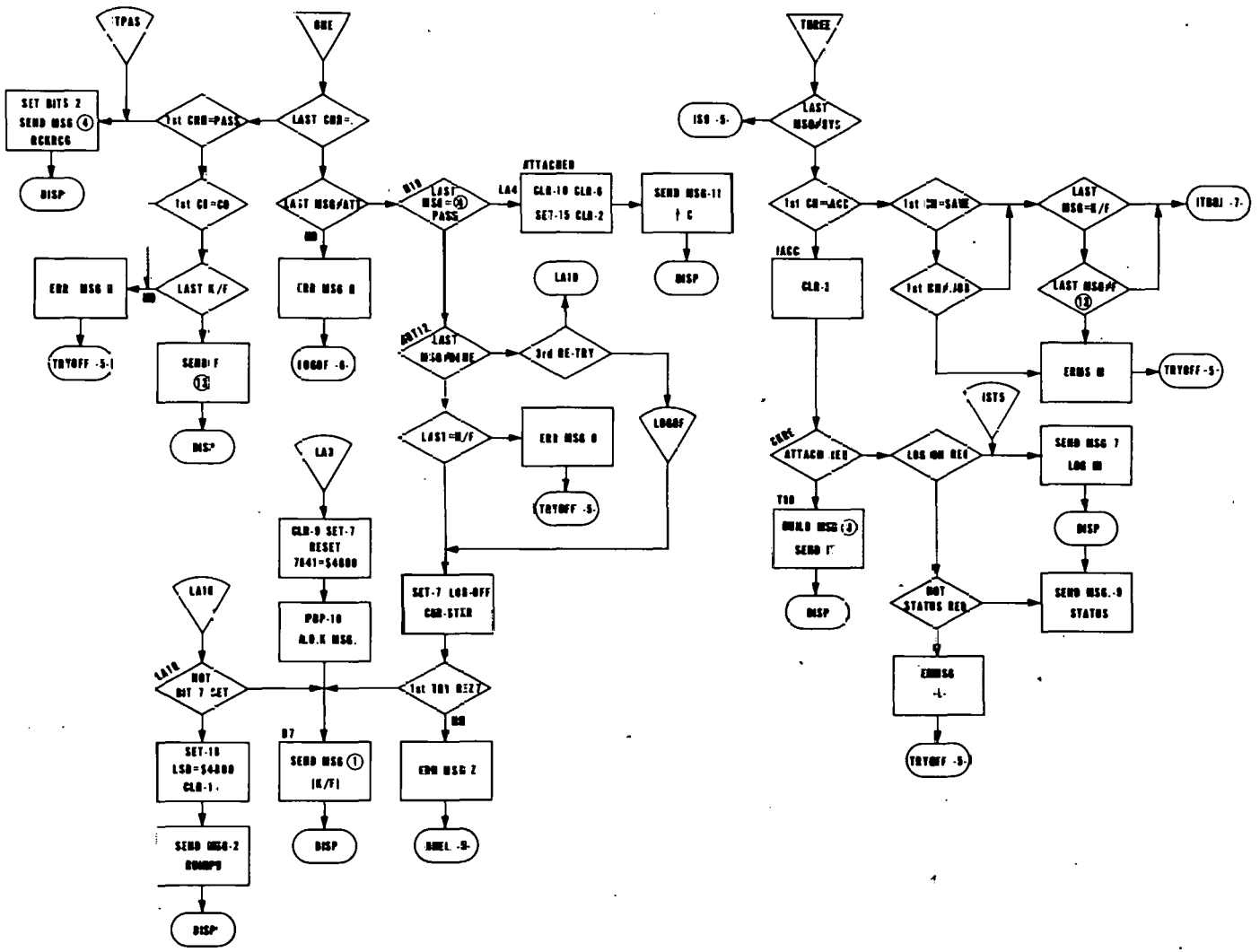


Fig. 21. DIAL subroutines.

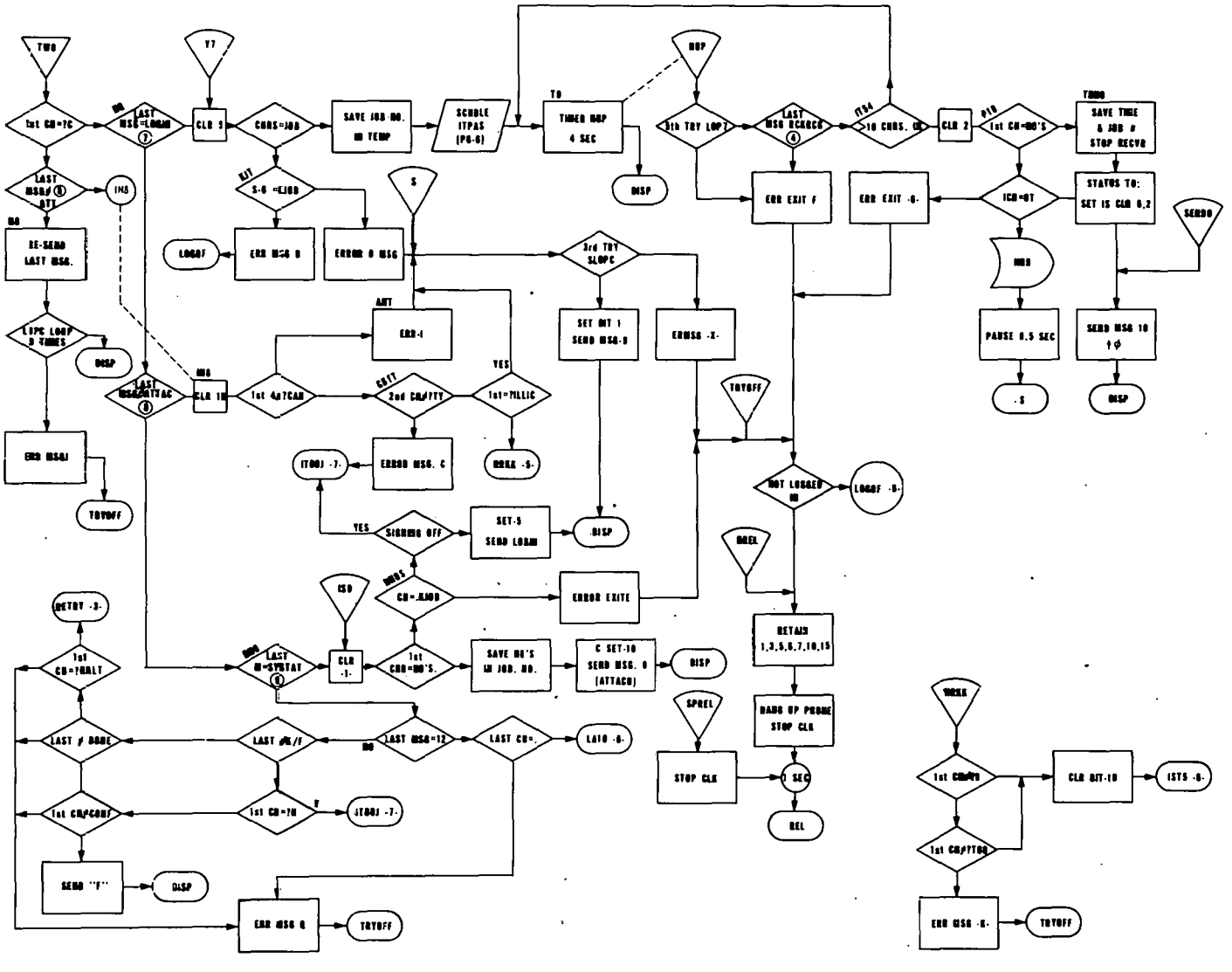


Fig. 22. DIAL subroutines.

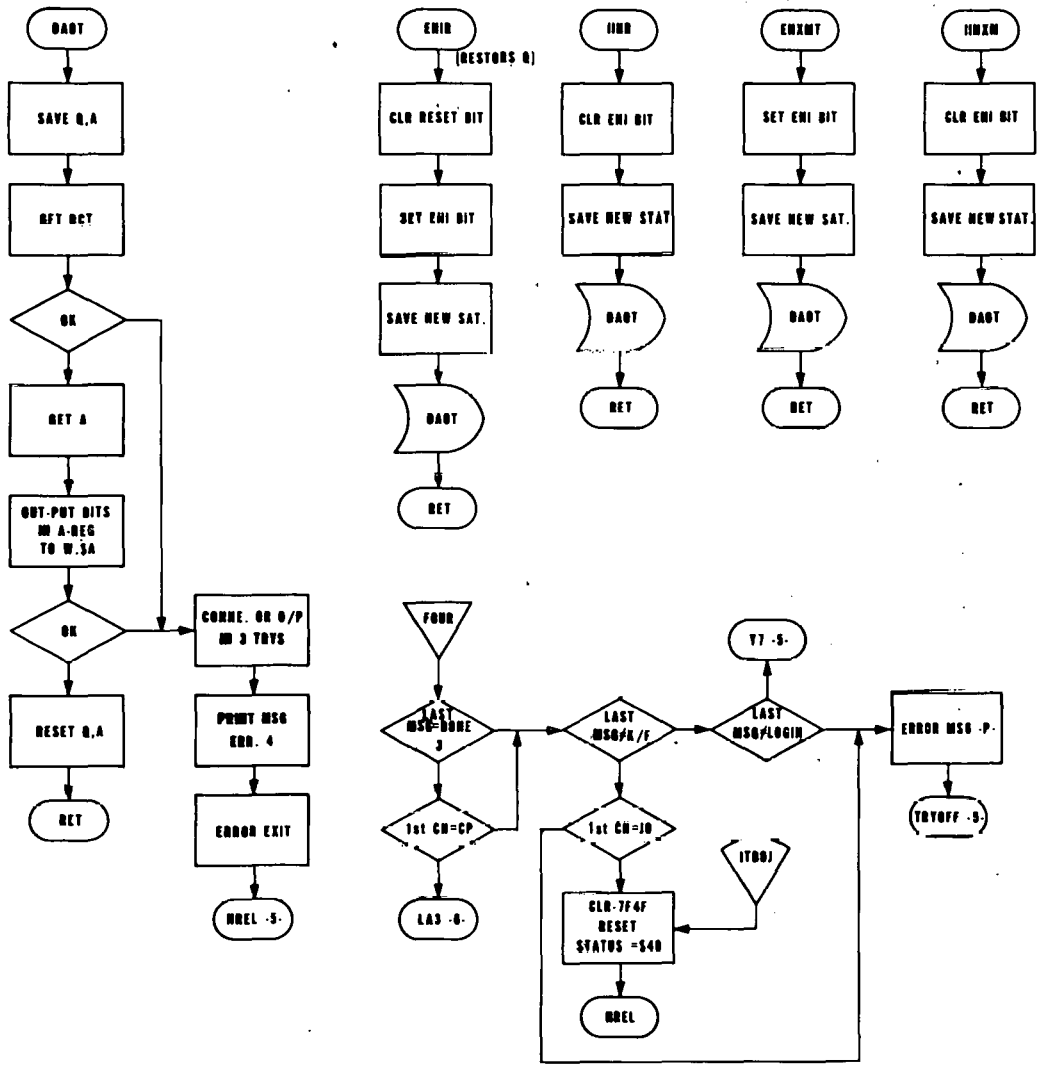


Fig. 23. DIAL subroutines.

is set and the disconnect procedure is used. If signing-off is in progress and response is normal, then "hang-up" the telephone is excuted.

8. When the CDC-1700 has sent RU INP, the received characters are scanned for special binary characters $2A and $19.. When received in series $2A,$19, the PDP-10 is ready for a string of 24 data words (one-half a log). If $2A is not followed by $19, a repeat of the last string is attempted. A maximum of 50 repeats per session is permitted before error message-T is typed. In addition to the scan for $2A and $19, the message is checked to determine if the first two characters "?HALT...etc" have been received. This would be an indication that the PDP-10 input program had stopped for some reason, and a restart by sending RU INP is attempted. After three unsuccessful attempts, "message-T" is typed and logging off is initiated.

9. Binary data blocks are read from buffer blocks on the drum ($C/$4800) as 16-bit words by subroutine STRIN (Fig. 19). The words are expanded into a block of forty-eight, 8-bit words, and a sum of all 48 words is added to the block as 8-bit words, 49 and 50. The subroutine XMTR is used to send the fifty, 8-bit word-string to the PDP-10. When the transmitter ready signal is set, an interrupt is generated on line 15 which executes XMTR for the next character. After all characters are sent, the receiver is enabled and a scan for the special binary characters is resumed as above.

   When STRIN has sent all words from the buffer, the DONE XX message is sent. XX represents the total number of logs that were sent

during the current session. If the PDP-10 does not agree with the log total, a "?HALT...etc" will cause a repeat.

10. Subroutine MM00 (Fig. 20) is used to send messages. An ASCII table of required messages is constructed with ALF commands (see ME1-ME13 in the listings). When included in the ADC table MEL, the message can be sent by entering the message number in the Q-register and jumping to MM00. The following messages may be sent:

Message No.	Message	Meaning
1	↑C↑C K/F	Sign-off
2	↑C↑C RU INP	Accept data
3	DONE	Data finished
4	RXXXX	Password
7	LOGI (6415,26)	Log-in
8	ATT XX[6415,26]	Attach
9	SYS[6415,26]	Status of jobs
10	↑ϕ	Suppress typing
11	↑C↑C	Go to monitor
12	R SET TTY	Run TTY set-up
13	F	Confirm sign-off

A standard error message routine is also simple to call (Fig. 20). The first part of the message is designed to type CALL KRYTER 3-1270 ERROR-. The caller loads the A-register with the desired ASCII character, less $30, and does a return jump to MSGG. The character will be printed following the word ERROR, and this is followed with a listing of the current core and common status words. A table of all the error messages that may be generated by both FRYLIN and DIAL is shown in the Appendix, Sect. 5.3.

REFERENCES

1. R. C. Gonzalez, D. N. Fry, and R. C. Kryter, "Results in the Application of Pattern Recognition Methods to Nuclear Reactor Core Component Surveillance," *IEEE Trans. Nucl. Sci.* 21(1), (February 1974).

2. D. N. Fry, "Experience in Reactor Malfunction Diagnosis Using On-line Noise Analysis," *Nucl. Technol.* 10, 273 (March 1971).

3. J. B. Bullock, G. R. Owens, and W. H. Sides, Jr., *Reactor On-line Computer Control Development at the HFIR, Vol. 2: Program Listings, Summaries and Logic Diagrams*, ORNL/TM-3679, Vol. 2 (February 1972).

4. J. B. Bullock and H. P. Danforth, *Reactor On-line Computer Control Development at the HFIR, Vol. 1: Objectives, System Design, Operating Experience and Safety Considerations*, ORNL-TM-3679, Vol. 1 (October 1972).

5. APPENDIX

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

5.1 Program Listings

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

```
001.                        NAM  FRYLIN  3/26/74
002.                  *
003.                  *
004.                  * WILL TRY AGAIN IF MAN IS IN USE ALSØ LØØKS
005.                   * FØR MANUAL USE ØF RØDS. IF FØUND,STARTS ØVER.
006.        7F4E              EQU  SYSTAT($7F4E),CØRSTA($6E),D($EA)
            006E
            00EA
007.                          EXT  MANBSY,AUTBSY,FRYLIN,RDPT,CØNECT
008.                           EXT  PAMLST,DRI72,Q8QFLT
009.                          EXT  DIALUP
010.        0088              EQU  ARE($88),ØNB($23),ZR($33)
            0023
            0033
011.        0089              EQU  AHEX($89),DECØ($A3)
            00A3
012.        7F05              EQU  TIM($7F05),INA($7F00),LAS($7F42)
            7F00
            7F42
013.        7F41              EQU  FRY($7F41),SYSLIB($EB)
            00EB



015. P0000 C400   FRYØN LDA  FRY  CK FØR REASØNABLE LSB'S.
     P0001 7F41
016. P0002 0133          SAM  BUFE-*-1  BAD.
017. P0003 9000          SUB  =N$6800  > $6800?
     P0004 6800
018. P0005 0134          SAM  ALRIG  ØK. (CAN GØ TØ $7FFF SAFELY,
019. P0006 0A25   BUFE   ENA  $25    BUT SEND MSG. NØW FØR ACTIØN.)
020. P0007 5800          RTJ  MSGG  MSG ---U
     P0008 011C
021. P0009 1856          JMP* REL
022. P000A C400   ALRIG LDA  INA  SCAN ØN?
     P000B 7F00
023. P000C 0113          SAN  GØ-*-1
024. P000D C808   KK2    LDA* REPRAT
025. P000E 0F41          ARS  1
026. P000F 186B          JMP* CALL
027. P0010 C400 X GØ     LDA  MANBSY
     P0011 7FFF X
028. P0012 0103          SAZ  NØ-*-1
029. P0013 0A03          ENA  3  FØR 4 MIN CALL
030. P0014 1866          JMP* CALL
031. P0015 00B3   REPRAT NUM  $B3  FØR 3 HR. CYCLE.
032. P0016 C400   NØ     LDA  TIM
     P0017 7F05
033. P0018 9400          SUB  LAS
     P0019 7F42
034. P001A 0121          SAP  CN-*-1
035. P001B 0918          INA  $18  ADD 1 DAY.
036. P001C 2000   CN     MUI  =N60  HRS TØ MINS.
     P001D 003C
037. P001E 98F6          SUB* REPRAT
```

```
038. P001F 0121          SAP  YER-*-1  BEEN LØNGER THAN 1 PERIØD?
 039. P0020 13EC          JMP* KK2
040. P0021 0C02   YER    ENQ  2
041. P0022 5C4C          RTJ* (D72)   M/C THE 1572.
042. P0023 FFFE          NUM  -1,0,0
     P0024 0000
     P0025 0000
043. P0026 0162          SQP  RU-*-1
044. P0027 0A03          ENA  3  FØR 4 MIN CALL
045. P0028 1852          JMP* CALL
046. P0029 0C00   RU     ENQ  0
047. P002A 0A0B          ENA  $B  READ HI LVL CH B FØR D.C. VØLTAGE
 048. P002B 5400 X        RTJ  RDPT
     P002C 7FFF X
049. P002D 980E          SUB* MIN  TØØ LØW?
050. P002E 0134          SAM  ER-*-1
051. P002F 880C          ADD* MIN  TØØ HI?
052. P0030 980C          SUB* MAX
053. P0031 0121          SAP  ER-*-1
054. P0032 180C          JMP* GØØN
055. P0033 D80A   ER     RAØ* THRE
056. P0034 C809          LDA* THRE
057. P0035 0121          SAP  ALA-*-1
058. P0036 18F2          JMP* RU
059. P0037 0A1E   ALA    ENA  $1E  MSGG --N
060. P0038 5800          RTJ  MSGG
     P0039 00EB
061. P003A 18D2          JMP* KK2  CALL BACK IN DT/2
062. P003B 8000   MIN    NUM  $8000
063. P003C 7FFF   MAX    NUM  $7FFF
064. P003D FFFC   THRE   NUM  -3
065. P003E 0A01   GØØN   ENA  1 SET IT BUSY NØW.
066. P003F 6400 X        STA  AUTBSY
     P0040 7FFF X
067. P0041 C000          LDA  =N$62B3  SET TIMER FØR 125 SEC. WRAPARØUND.
     P0042 62B3
068. P0043 682C          STA* MØRE
069. P0044 C000 X         LDA  =XPAMLST
     P0045 7FFF X
070. P0046 09FA           INA  -5       INDEX TØ BUFCTR IN MUXBUF.
 071. P0047 60FF           STA- I
072. P0048 0C00          ENQ  0
073. P0049 CA59   LP     LDA* MUXNØM,Q TXFR TABLE TØ PAMLST IN MUXBUF.
 074. P004A 6301          STA- 1,B
075. P004B 0814          TRQ  A
076. P004C 09F4           INA  -11
077. P004D 0102          SAZ  DN-*-1
078. P004E 0D01          INQ  1
079. P004F 18F9          JMP* LP
080. P0050 0C03   DN     ENQ  3  FØR DIV.
081. P0051 C000           LDA  =N$D40    CØMPUTE  REG. FØR DESIRED SCANRA
     P0052 0D40
082. P0053 3000          DVI  =N128  = SCANRATE IN PPS.
     P0054 0080
083. P0055 6805           STA* REG72
```

```
084. P0056 0C02           ENQ  2
085. P0057 5C17           RTJ* (D72)
086. P0058 0001           NUM  1
087. P0059 7FFF X         ADC  CØNECT
088. P005A 0000   REG72   NUM  0
089. P005B 0171         SQM  NUPS-*-1
090. P005C 1806           JMP* UM
091. P005D 0A01   NUPS    ENA  1  2 MIN CALL.
092. P005E 181C           JMP* CALL
093.              REL     RELEAS (FRYØN-REL-1),T,X
093. P005F 54F4
093. P0060 1901
093. P0061 FF9F
094. P0062 5488   UM      RTJ- (ARE)
095. P0063 0046           ADC  $46  WØRD $A  FØR SB SWITCH MØVEMENT.
096. P0064 A02E           AND- ØNB+11  BIT 11
097. P0065 010A           SAZ  SØ-*-1  SKIP IF SB SW IS NØT IN ØFF.
098. P0066 C809           LDA* MØRE
099. P0067 0139           SAM  ADN-*-1
100.              T2      TIMER UM-T2-1,5,X,0  BASIC UNITS. LVL-5
100. P0068 54F4
100. P0069 1105
100. P006A 7FF8
101. P006B 0000           NUM  0
102. P006C D803           RAØ* MØRE
103. P006D 14EA           JMP- ($EA)
104. P006E 7FFF X D72     ADC  DRI72
105. P006F 0000   MØRE    NUM  0
106. P0070 18B8   SØ      JMP* RU
107. P0071 C400 X ADN     LDA  AUTBSY  IF AUTBSY IS NØW CLEAR,
     P0072 0040 X
108. P0073 0111           SAN  ØK-*-1  IT'S PRØBABLY BECAUSE ØF
109. P0074 18B4           JMP* RU  MUXBUF ØVERANGE, SØ RE-START.
110. P0075 C400   ØK      LDA  TIM
     P0076 7F05
111. P0077 6400           STA  LAS
     P0078 7F42
112. P0079 C89B           LDA* REPRAT  CALL FØR NEXT FRYLIN.
113. P007A 6804   CALL    STA* DTRP
114.                      TIMER (FRYLIN),4,0,3
114. P007B 54F4
114. P007C 1034
114. P007D FFFF X
115. P007E 0000   DTRP    NUM  0
116. P007F C400           LDA  FRY  ANY LØGS?
     P0080 7F41
117. P0081 9000           SUB  =N$4800
     P0082 4800
118. P0083 010D           SAZ  H-*-1
119. P0084 C400           LDA  INA  SCAN ON?
     P0085 7F00
120. P0086 0109           SAZ  JLIA-*-1
121. P0087 C400           LDA  TIM   TIME RIGHT?
     P0088 7F05
122. P0089 09F8           INA  -7
```

```
123. P008A 0126          SAP  H-*-1
124. P008B C400          LDA  FRY   # LØGS >8?
     P008C 7F41
125. P008D 9000          SUB  =N$4950
     P008E 4950
126. P008F 0131          SAM  H-*-1
127. P0090 181E    JLIA  JMP* LIAD
128. P0091 C400    H     LDA  SYSTAT
     P0092 7F4E
129. P0093 0FC8          ALS  8  SIGING ØFF BIT SET?
130. P0094 0121          SAP  NSET-*-1
131. P0095 1803          JMP* JMLI
132. P0096 0FC1    NSET  ALS  1 SIGNED ØFF?
133. P0097 0137          SAM  ØFF-*-1
134. P0098 0C07    JMLI  ENQ  7
135. P0099 5800          RTJ  SETST
     P009A 0079
136. P009B E806          LDQ* DIALAD
137. P009C 0D01          INQ  1
138. P009D C6EB          LDA- (SYSLIB),Q  DIALUP ALREADY RUNNING?
139. P009E 0101           SAZ  ATT-*-1
140. P009F 18BF    ØFF   JMP* REL
141. P00A0 1850    ATT   JMP* K GØ DØ STATUS WITH ØFF BIT SET.
142. P00A1 7FFF X DIALAD ADC  DIALUP
143. P00A2 0001    MUXNØM NUM  1,0,$B,0,7,$20,$80,$7D,$500,0,0,$A0
     P00A3 0000
     P00A4 000B
     P00A5 0000
     P00A6 0007
     P00A7 0020
     P00A8 0080
     P00A9 007D
     P00AA 0500
     P00AB 0000
     P00AC 0000
     P00AD 00A0
144.               * THE ABØVE IS PAMLST PRESETS FØR STANDARD CASE.
145.                *  STARTING WITH BUFCTR AND ENDING WITH NUMBUF.
146.                ************************************************
147. P00AE E8F2    LIAD  LDQ* DIALAD
148. P00AF 0D01          INQ  1
149. P00B0 C6EB          LDA- (SYSLIB),Q SEE IF DIALUP NØW RUNNING.
150. P00B1 0101           SAZ  RUNNØW-*-1
151. P00B2 18AC          JMP* REL
152. P00B3 C400    RUNNØW LDA  SYSTAT+1  JØB # (IN ASCII)
     P00B4 7F4F
153. P00B5 010A          SAZ  STJØN-*-1
154. P00B6 B000          EØR  =N$3030
     P00B7 3030
155. P00B8 0C00          ENQ  0
156. P00B9 0FE8          LLS  8
157. P00BA 0FC4          ALS  4
158. P00BB 0F6C          LRS  12  NØW DECIMAL IN A REG.
159. P00BC 54A3          RTJ- (DECØ)
160. P00BD 0000          NUM  0,0,0
     P00BE 0000
     P00BF 0000
161. P00C0 682B    STJØN STA* JB  SAVE #
```

```
162. P00C1 0822          TRA  Q  HØLD IN Q
163. P00C2 C06E          LDA- CØRSTA
164. P00C3 0119          SAN  SØM-*-1
165. P00C4 C400          LDA  SYSTAT  PRØBABLE 1700 BØMB.
     P00C5 7F4E
166. P00C6 A032          AND- ØNB+15  SAVE ØNLY BIT 15 FRØM ØLD
167. P00C7 606E           STA- CØRSTA  STATUS ,SINCE IT HAS BØMBED.
168. P00C8 C06E   MØS     LDA- CØRSTA  CØRE STATUS WØRD.
169. P00C9 6400          STA  SYSTAT  NØRMALIZE THE TWØ.
     P00CA 7F4E
170. P00CB 0129          SAP  NSØMC-*-1
171. P00CC 1820          JMP* SØMC
172. P00CD 9400   SØM    SUB  SYSTAT  IS LØ CØRE= CØMM STATUS.
     P00CE 7F4E
173. P00CF 0103          SAZ  UHU1-*-1
174. P00D0 0A08          ENA  8  STATUS MISMATCH ERR.
175. P00D1 5853          RTJ* MSGG   --8
176. P00D2 18F5          JMP* MØS  ASSUME CØRE IS GØØD,TRY AGN.
177. P00D3 C06E   UHU1   LDA- CØRSTA  LØGGED IN?
178. P00D4 0137          SAM  MRER-*-1
179. P00D5 0141   NSØMC  SQZ  NØLJ-*-1
180. P00D6 1822          JMP* JBNØ
181. P00D7 0C05   NØLJ   ENQ  5  LØGIN REQUEST.
182. P00D8 583B          RTJ* SETST
183. P00D9 0C0A          ENQ  10 CLR 10
184. P00DA 5842          RTJ* CLRS
185. P00DB 1829          JMP* CLRIØ
186. P00DC 0151   MRER   SQN  MAYB-*-1
187. P00DD 1811          JMP* LØSTJ
188. P00DE 0161   MAYB   SQP  MAY2-*-1
189. P00DF 1820          JMP* BDJ  BAD
190. P00E0 0DC3   MAY2   INQ  -60  <60?
191. P00E1 0171          SQM  RER-*-1  YES.
192. P00E2 181D          JMP* BDJ  BAD.
193. P00E3 C06E   RER    LDA- CØRSTA
194. P00E4 A02A          AND- ØNB+7
195. P00E5 0114          SAN  GØTEL-*-1
196. P00E6 0A09          ENA  9  RE-ENTRY ERR MSG.
197. P00E7 583D          RTJ* MSGG  --9
198. P00E8 1800          JMP  REL
     P00E9 FF75
199. P00EA 1818   GØTEL   JMP* DI
200. P00EB 0000   JB      NUM  0
201. P00EC E8FE   SØMC    LDQ* JB
202. P00ED 015A          SQN  JBNØ-*-1
203. P00EE 0A02   LØSTJ   ENA  2  LØST JØB - MSG.
204. P00EF 5835          RTJ* MSGG ---2
205. P00F0 0C01   K       ENQ  1  GØ GET STATUS BIT.
206. P00F1 5800          RTJ  SETST
     P00F2 0021
207. P00F3 0C05          ENQ  5  CLR 5
208. P00F4 5828          RTJ* CLRS
209. P00F5 0C0A          ENQ  10 AND 10
210. P00F6 5826          RTJ* CLRS
211. P00F7 180F          JMP* TIMER
212. P00F8 0176   JBNØ   SQM  BDJ-*-1
213. P00F9 0DC3          INQ  -60  SEE IF JØB # <60 &>0.
214. P00FA 0164          SQP  BDJ-*-1
```

```
215. P00FB 0136          SAM  DI-*-1  SKP IF LØGGED IN WITH GØØD#.
 216. P00FC 0A07          ENA  7
217. P00FD 5827          RTJ* MSGG --7
218. P00FE 18F1          JMP* K  GØ GET STATUS.
219. P00FF 0A11    BDJ   ENA  $11  BAD JØB # MSG--A
220. P0100 5824          RTJ* MSGG  --11
221. P0101 18EE          JMP* K  TRY GETTING STATUS.
222. P0102 0C0A    DI    ENQ  10  SET ATTACHING BIT AND GØ DIAL.
 223. P0103 5810          RTJ* SETST
224. P0104 0C01    CLRIØ ENQ  1  CLR 1 ALSØ
225. P0105 5817          RTJ* CLRS
226. P0106 C400    TIMER  LDA  SYSTAT
     P0107 7F4E
227. P0108 A000          AND  =N$4E2  CLR ALL XCPT 10,7,6,5,1
     P0109 04E2
228. P010A 6400          STA  SYSTAT
     P010B 7F4E
229. P010C 606E          STA- CØRSTA
230.                     TIMER (DIALUP),4,0,3
230. P010D 54F4
230. P010E 1034
230. P010F 80A1 X
231. P0110 0002          NUM  2
232. P0111 1800    YES   JMP  REL
     P0112 FF4C
233. P0113 0B00    SETST  NØP  0  TØ SET BIT NØ. IN Q REG.
234. P0114 C400          LDA  SYSTAT
     P0115 7F4E
235. P0116 A233          AND- ZR,Q
236. P0117 B223          EØR- ØNB,Q
237. P0118 6400          STA  SYSTAT
     P0119 7F4E
238. P011A 606E          STA- CØRSTA
239. P011B 1CF7          JMP* (SETST)
240. P011C 0B00    CLRS  NØP  0  TØ CLR STATUS BIT SET IN Q REG.
 241. P011D C400          LDA  SYSTAT
     P011E 7F4E
242. P011F A233          AND- ZR,Q
243. P0120 6400          STA  SYSTAT
     P0121 7F4E
244. P0122 606E          STA- CØRSTA
245. P0123 1CF8          JMP* (CLRS)
246.                   * ERRØR MSG. RØUTINE. ENTER WITH MSG NØ.IN Q REG.
 247. P0124 0000   MSGG  NUM  0  BUFFER 5 MSG REPEATS & RETURN.
 248. P0125 882E          ADD* ASCH  MAKE IT LEGAL ASCII.
249. P0126 6824          STA* M+12
250. P0127 0AFA          ENA  -5
251. P0128 682A          STA* CTLP
252. P0129 C400          LDA  SYSTAT
     P012A 7F4E
253. P012B 5489          RTJ- (AHEX)
254. P012C 8020          ADC  (M+14-*)
255. P012D C06E          LDA- CØRSTA
256. P012E 5489          RTJ- (AHEX)
257. P012F 8020          ADC  (M+17-*)
258.                   MSGK  FWRITE $E,MADD-MSGK-1,M-MSGK-1,20,A,4,4,,X
 258. P0130 54F4
```

```
258. P0131 0D44
258. P0132 0007
     P0133 0000
258. P0134 100E
258. P0135 0014
     P0136 000D
259. P0137 14EA          JMP- (D)
260. P0138 C81A    MADD  LDA* CTLP
261. P0139 0122          SAP  RE7-*-1  SKIP IF TYPED 5 TIMES.
262. P013A D818          RAØ* CTLP
263. P013B 18F4          JMP* MSGK
264. P013C 1CE7    RE7    JMP* (MSGG)
265. P013D 0000          NUM  0
266. P013E 4341    M     ALF  20,CALL KRYTER 3-1270 ERRØR 1 $     $
     P013F 4C4C
     P0140 204B
     P0141 5259
     P0142 5445
     P0143 5220
     P0144 332D
     P0145 3132
     P0146 3730
     P0147 2045
     P0148 5252
     P0149 4F52
     P014A 2031
     P014B 2024
     P014C 2020
     P014D 2020
     P014E 2024
     P014F 2020
     P0150 2020
     P0151 2020
267. P0152 FFFA    CTLP  NUM  -5
268. P0153 2030    ASCH   NUM  $2030
269.                     END
```

```
I        00FF  SYSTAT   7F4E  CØRSTA   006E  D        00EA  ARE      0088
 ØNB      0023  ZR       0033  AHEX     0089  DECØ     00A3  TIM      7F05
 INA      7F00  LAS      7F42  FRY      7F41  SYSLIB   00EB  FRYØN    0000P
 BUFE     0006P ALRIG    000AP KK2      000DP GØ       0010P REPRAT   0015P
 NØ       0016P CN       001CP YER      0021P RU       0029P ER       0033P
 ALA      0037P MIN      003BP MAX      003CP THRE     003DP GØØN     003EP
 LP       0049P DN       0050P REG72    005AP NUPS     005DP REL      005FP
 UM       0062P T2       0068P D72      006EP MØRE     006FP SØ       0070P
 ADN      0071P ØK       0075P CALL     007AP DTRP     007EP JLIA     0090P
 H        0091P NSET     0096P JMLI     0098P ØFF      009FP ATT      00AOP
 DIALAD   00A1P MUXNØM   00A2P LIAD     00AEP RUNNØW   00B3P STJØN    00COP
 MØS      00C8P SØM      00CDP UHU1     00D3P NSØMC    00D5P NØLJ     00D7P
 MRER     00DCP MAYB     00DEP MAY2     00EOP RER      00E3P GØTEL    00EAP
 JB       00EBP SØMC     00ECP LØSTJ    00EEP K        00FOP JBNØ     00F8P
 BDJ      00FFP DI       0102P CLRIØ    0104P TIMER    0106P YES      0111P
 SETST    0113P CLRS     011CP MSGG     0124P MSGK     0130P MADD     0138P
 RE7      013CP M        013EP CTLP     0152P ASCH     0153P DIALUP   010FX
 QSQFLI   7FFFX DRI72    006EX PAMLST   0045X CØNECT   0059X RDPT     002CX
 FRYLIN   007DX AUTBSY   0072X MANBSY   0011X
J
```

```
001.                        NAM DIAL  3/27/74  TØ CALL PDP-10
002.        7F4E            EQU  SYSTAT($7F4E),CØRSTA($6E),D($EA)
             006E
            00EA
003.                        EXT  DIALUP
004.        00EB            EQU  SYSLIB($EB)
005.        0064             EQU  INBUF(100),BUF(24),FRY($7F41),AHEX($89)
             0018
            7F41
            0089
006.        0088            EQU  ARE($88),ØNB($23),ZR($33),TIM($7F05),INA($7
             0023
            0033
            7F05
            7F00
007. P0000 C8FE   START NUM  $C8FE,$8808,$6400,$13B,$C8FA,$8805,$6400,$1
      P0001 8808
     P0002 6400
     P0003 013B
     P0004 C8FA
     P0005 8805
     P0006 6400
     P0007 013F
     P0008 1803
008. P0009 0196             ADC  FETCH-START  THIS IS TØ PUT ADD. ØF FETCH &
 009. P000A 017E             ADC  XMTR-START  INTØ INTRRUPT TRAPS AT$13B&13F
 010. P000B C06E             LDA- CØRSTA
011. P000C 681A             STA* ENTRYS
012. P000D C819   DIAL    LDA* ENTRYS
013. P000E 6400           STA  SYSTAT  RESET STATUS TØ ENTRY VALUE.
      P000F 7F4E
014. P0010 0C03           ENQ  3  SET DIALING-BIT.
015. P0011 5800           RTJ  SETST
     P0012 0351
016. P0013 0A00           ENA  0
017. P0014 6800            STA  LAS
     P0015 0340
018. P0016 5800           RTJ  NULTA
     P0017 0290
019. P0018 C000           LDA  =N$7000 FØR Ø/H,D/A & D/T BITS ØF MØDEM.
      P0019 7000
020. P001A 584B           RTJ* DAØT
021. P001B 0C78           ENQ  120 DELAY 2 SECS.
022. P001C 583E           RTJ* DEL
023. P001D 5488           RTJ- (ARE)
024. P001E 0045           ADC  $45  WØRD 9
025. P001F A028           AND- ØNB+5 GET DIAL TØNE?
026. P0020 0116           SAN  ITS-*-1  YES
027. P0021 0A05           ENA  5  ERRØR 5
028. P0022 5800           RTJ  MSGG ERRØR- NØ DIAL TØNE.
     P0023 0301
029. P0024 1800           JMP  NREL
     P0025 0350
030. P0026 0000   ENTRYS NUM  0  STATUS ØAT ENTRY TIME.
031. P0027 C000   ITS     LDA  =N$5000  DRØP D/A DURING DIALING.
      P0028 5000
```

```
032. P0029 583C          RTJ* DAØT
033. P002A 0C02          ENQ  2  PAUSE
034. P002B 582F          RTJ* DEL
035. P002C C831          LDA* T6+1  INCREASE PRIØRITY FØR DIALING
 036. P002D 0905          INA  5     ACCURACY.
037. P002E 682F          STA* T6+1
038. P002F 0A03          ENA  3  READY TØ DIAL 3-1881
039. P0030 5817          RTJ* YE
040. P0031 0A01          ENA  1
041. P0032 5815          RTJ* YE
042. P0033 0A08          ENA  8
043. P0034 5813          RTJ* YE
044. P0035 0A08          ENA  8
045. P0036 5811          RTJ* YE
046. P0037 0A01          ENA  1
047. P0038 580F          RTJ* YE
048. P0039 C824          LDA* T6+1
049. P003A 09FA          INA  -5  RESTØRE PRIØRITY.
050. P003B 6822          STA* T6+1  TØ ØRIGINAL.
051. P003C C000          LDA  =N$7000  LAST NØ IS DIALED SØ BRING
      P003D 7000
052. P003E 6800          STA  ØUTST UP D/A,D/T&Ø/H. ALSØ SAVE
      P003F 0111
053. P0040 5825          RTJ* DAØT  CURRENT STATUS AS INITIAL.
 054. P0041 5800          RTJ  ENIR  ENABLE RECVR.
      P0042 010F
055. P0043 1800          JMP  NR3  GØ START CLØCK.
      P0044 01EA
056. P0045 FFFA   HI2    NUM  -5  FØR 5 DISCØNNECT TRYS.
057. P0046 0000   AS     NUM  0
058. P0047 0B00   YE     NØP  0
059. P0048 68FD          STA* AS
060. P0049 C8FC   DØNE   LDA* AS
061. P004A 09FE          INA  -1
062. P004B 0123          SAP  NØP-*-1
063. P004C 0C3C          ENQ  60  PAUSE 1 SEC.
064. P004D 580D          RTJ* DEL  BEFØR DIALING NEXT NØ.
065. P004E 1CF8          JMP* (YE)
066. P004F 68F6   NØP    STA* AS
067. P0050 C031          LDA-  $31  BRING DØWN Ø/H FØR 67 MSEC.
 068. P0051 5814          RTJ* DAØT    TØ GØVE DIAL PULSE.
069. P0052 0C03          ENQ  3
070. P0053 5807          RTJ* DEL
071. P0054 C000          LDA  =N$5000 NØW BACK UP FØR 33 MSEC.
      P0055 5000
072. P0056 580F          RTJ* DAØT
073. P0057 0C01          ENQ  1  33. MSEC
074. P0058 5802          RTJ* DEL
075. P0059 13EF          JMP* DØNE MØRE?
076. P005A 0000   DEL    NUM  0
077. P005B 4804          STQ* VALT  TIME DELAY GENERATØR LØØP.
 078.               T6    TIMER DAC-T6-1,5,X,0
078. P005C 54F4
078. P005D 1105
078. P005E 0007
```

```
175. P00CD 5800          RTJ  SETST
     P00CE 0295
176. P00CF 0C03          ENQ  3  FØR SENDING MSG.3
177. P00D0 0AC3          ENA  -60  FØR 1 MIN.
178. P00D1 6800          STA  TMDØT
     P00D2 01C3
179. P00D3 1800          JMP  MMØØ  TØ TELL PDP-10 TØ SAVE DAILY
     P00D4 043B
180. P00D5 0A26    ERV   ENA  $26
181. P00D6 5800          RTJ  MSGG  -V
     P00D7 024D
182. P00D8 1800          JMP  TRYØFF
     P00D9 02FB
183. P00DA 0C00    ØRR   ENQ  0  SET BIT 0 AND STXR BUSY FLG.
184. P00DB 5800           RTJ  SETST
     P00DC 0287
185. P00DD E03F          LDQ- $3F CØDE FØR XFRING DATA.
186. P00DE 482E          STQ* STXR  SET BUSY.
187.               RD    READ 5,DRMWR-RD-1,TAB-RD-1,BUF,B,5,5,,X
187. P00DF 54F4
187. P00E0 0355
187. P00E1 000B
     P00E2 0000
187. P00E3 0005
187. P00E4 0018
     P00E5 002F
188. P00E6 000C          NUM  $C  MSB ØF DATA.
189. P00E7 4800    LSB   NUM  $4800  INITIAL LSB ØF DATA.
190. P00E8 0161          SQP  DRM-*-1
191. P00E9 18C8          JMP* STRIN  READY WITH NEXT SRING.
192. P00EA 14EA    DRM   JMP- (D)  DISP
193. P00EB C8FB    DRMWR LDA* LSB
194. P00EC 0918          INA  BUF  FØR NEXT READ.
195. P00ED 68F9          STA* LSB
196. P00EE 0A00    EXP   ENA  0
197. P00EF 681F          STA* CKSUM
198. P00F0 0C18          ENQ  RIIF
199. P00F1 40FF          STQ- I
200. P00F2 0C17          ENQ  BUF-1
201. P00F3 CA1C    TAL2  LDA* TAB,Q  * THIS LØØP CØNVERTS TAB INTØ
202. P00F4 A00A           AND- 10 =FF * A TABLE ØF 8 BIT WDS. WITH
203. P00F5 6B1A           STA* TAB,B  * A CHECK SUM WØRD AS LAST WD.
204. P00F6 8818           ADD* CKSUM
205. P00F7 6817          STA* CKSUM
206. P00F8 C0FF          LDA- I
207. P00F9 09FE          INA  -1
208. P00FA 60FF          STA- I
209. P00FB CA14          LDA* TAB,Q
210. P00FC 0F48          ARS  8
211. P00FD A00A          AND- 10 =FF
212. P00FE 6B11          STA* TAB,B
213. P00FF 880F          ADD* CKSUM
214. P0100 680E          STA* CKSUM
215. P0101 0DFE          INQ  -1
216. P0102 0171          SQM  DØR-*-1
```

```
130. P0096 6800          STA  STPC  ALLØW CLK TØ START.
     P0097 01FF
131. P0098 1800           JMP DIAL TRY RE:DIALING.
     P0099 FF73
132. P009A 0A29    ERMY   ENA  $29
133. P009B 5800           RTJ  MSGG   MSG---Y
     P009C 0288
134. P009D 0C07           ENQ  7  SET SIGN ØFF.
135. P009E 5800           RTJ  SETST
     P009F 02C4
136. P00A0 18F4           JMP* STCRK  GØ START CLK.
137. P00A1 FFDD    AR     NUM  -34  CTR FØR ERR MSG.
138. P00A2 C8FE    LAT    LDA* AR
139. P00A3 0128           SAP  MSGT-*-1  BEEN TRYING FØR 5 HRS. SØ ERRØR.
140. P00A4 D8FC            RAØ* AR
141. P00A5 0AFA           ENA  -5
142. P00A6 68E0           STA* HI  RESTE
143. P00A7 E011           LDQ- $11 =7FFF
144. P00A8 0A00           ENA  0
145. P00A9 58BB           RTJ* DAØT  HANG UP .
146. P00AA 58AF           RTJ* DEL PAUSE 9.1 MINS.
147. P00AB 18E3           JMP* DISC
148. P00AC 0A01    MSGT   ENA  1
149. P00AD 5800           RTJ  MSGG  ERRØR -NØ ANSW IN 5 HRS.
     P00AE 0276
150. P00AF 1800    INREL  JMP  NREL
     P00B0 02C5
151. P00B1 0000    FI     NUM  0
152. P00B2 0ACD    STRIN  ENA  -50  RESET LØØP CTR.
153. P00B3 68FD           STA* FI
154. P00B4 C858    STR2   LDA* STXR  STRING BUSY.?
155. P00B5 0107           SAZ  NØTBS-*-1
156. P00B6 0C3B           ENQ  59  FØR 1 SEC.
157. P00B7 58A2           RTJ* DEL
158. P00B8 D8F8            RAØ* FI
159. P00B9 C8F7           LDA* FI
160. P00BA 0131           SAM  NERV1-*-1
161. P00BB 181A           JMP* ERV
162. P00BC 18F7    NERV1  JMP* STR2
163. P00BD 0C0C    NØTBS  ENQ  12
164. P00BE 5800           RTJ  SETST
     P00BF 02A4
165. P00C0 0C0D           ENQ  13
166. P00C1 5800           RTJ  CLRS
     P00C2 02AA
167. P00C3 C824           LDA* LSB
168. P00C4 9400           SUB  FRY
     P00C5 7F41
169. P00C6 0917           INA  BUF-1
170. P00C7 0121           SAP  NØRR1-*-1
171. P00C8 1812           JMP* ØRR
172. P00C9 0C0C    NØRR1  ENQ  12  SHØW THAT,
173. P00CA 5800           RTJ  CLRS  XFR IS FINISHED.
     P00CB 02A1
174. P00CC 0C09           ENQ  9  SET FØR SAVING DAILY DATA
```

```
079. P005F 0000   VALT   NUM  0
080. P0060 14EA         JMP- (D)
081. P0061 0000   DC     NUM  0
082. P0062 0000   QSV    NUM  0
083. P0063 0000   ASV    NUM  0
084. P0064 1CF5   BAC    JMP* (DEL)  RETURN.
085. P0065 0000   DAØT   NUM  0  THIS IS DRIVER FØR DIG-ØUTS ØN
086. P0066 48FB          STQ* QSV  WØRD $A. ENTERED WITH PATTERN IN
087. P0067 0C03          ENQ  3  A REG. THAT IS TØ BE ØUT-PUTTED.
088. P0068 48F8          STQ* DC
089. P0069 68F9         STA* ASV
090. P006A E02D   DCTØN LDQ- $2D  DCT CØNNECT CØDE.
091. P006B 0500         IIN  0
092. P006C 0202         INP  2
093. P006D 1807         JMP* DCTØK  GØ IT.
094. P006E 0B00         NØP  0
095. P006F E8F1         LDQ* DC
096. P0070 0DFE         INQ  -1
097. P0071 48EF         STQ* DC
098. P0072 017E         SQM  DCER-*-1
099. P0073 18F6         JMP* DCTØN
100. P0074 E000   DCTØK LDQ  =N$900A  FØR WØRD A
     P0075 900A
101. P0076 C8EC         LDA* ASV
102. P0077 0304         ØUT  4
103. P0078 E8E9         LDQ* QSV
104. P0079 0400        EIN  0
105. P007A 1CEA         JMP* (DAØT)  ØK SØ RETURN.
106. P007B 0B00         NØP  0
107. P007C E8E4         LDQ* DC
108. P007D 0143         SQZ  DCER-*-1  ERRØR
109. P007E 0DFE         INQ  -1
110. P007F 48E1         STQ* DC
111. P0080 18F3         JMP* DCTØK
112. P0081 0A04   DCER   ENA  4
113. P0082 0400         EIN  0
114. P0083 5800         RTJ  MSGG  DCT ØR DAC FAILED ØN ØUT:PUT.
     P0084 02A0
115. P0085 1300         JMP  NREL
     P0086 02EF
116. P0087 FFFA   HI     NUM  -5 TRY TØ RE-DIAL 5 TIMES.
117. P0088 0A00   NØT     ENA  0
118. P0089 6800         STA  STPC  STØP CLØCK
     P008A 020C
119. P008B C8FB         LDA* HI  TRIED 5 TIMES?
120. P008C 0131         SAM  NLAT-*-1
121. P008D 1815         JMP* LAT  GIVE UP FØR NØW.
122. P008E D8F8   NLAT   RAØ* HI
123. P008F 0A00   DISC   ENA  0  DIS-CØNNECT PHØNE LINE.
124. P0090 58D4         RTJ* DAØT
125. P0091 E030         LDQ- $30  133 SEC PAUSE.
126. P0092 58C7         RTJ* DEL
127. P0093 C8B1         LDA* HI2
128. P0094 0125         SAP  ERMY-*-1
129. P0095 0AFE   STCRK ENA  -1
```

```
217. P0103 18EF          JMP* TAL2
218. P0104 0F48   DØR    ARS  8 GET MSB ØF CK SUM
219. P0105 683A          STA* TAB+2*BUF  =CKSUM, SØ STØRE AS LAST WD.
220. P0106 C808           LDA* CKSUM
221. P0107 A00A          AND- 10
222. P0108 6838          STA* TAB+1+2*BUF  LSB ØF CKSUM
223. P0109 0A00          ENA  0
224. P010A 6803          STA* PTR
225. P010B 1873          JMP* XMTR
226. P010C 0000   STXR   NUM  0  STRING XFR IN PRØGRESS FLG.
227. P010D 0000   PTR    NUM  0   STRING PØINTER.
228. P010E 0000   CKSUM  NUM  0
229. P010F 0041          BZS  TAB(2*BUF+17)
230. P0150 0000   ØUTST  NUM  0  ØUT-PUT STATUS & DATA WØRD.
231. P0151 0000   ENIR   NUM  0  THIS LØØP WILL ENABLE RECVR. INTER.
232. P0152 C8FD           LDA* ØUTST  AND CLR THE RESET BIT.
233. P0153 A03E          AND- ZR+11  CLR. BIT IN CASE IT'S SET.
234. P0154 B02E           EØR- ØNB+11  NØW SET IT.
235. P0155 0B00          NØP  0
236. P0156 A03D          AND- ZR+10  CLR THE RESET BIT.
237. P0157 68F8          STA* ØUTST  RESTØRE STATUS WØRD.
238. P0158 5822          RTJ* DAØTT
239. P0159 1CF7          JMP* (ENIR)
240. P015A 0000   IIHR   NUM  0  THIS LØØP WILL INHIBIT RECVR.
241. P015B C8F4           LDA* ØUTST  INTERRUPTS.
242. P015C A03E          AND- ZR+11  CLEAR IT.
243. P015D 68F2          STA* ØUTST
244. P015E 581C          RTJ* DAØTT
245. P015F 1CFA          JMP* (IIHR)
246. P0160 0000   ENXMT  NUM  0  LØØP TØ ENABLE XMTR. INTERRUPTS.
247. P0161 C8EE           LDA* ØUTST
248. P0162 A03C          AND- ZR+9   1 ST CLR IT.
249. P0163 B02C          EØR- ØNB+9  NØW SET IT.
250. P0164 68EB          STA* ØUTST  RESTØRE STATUS.
251. P0165 5815          RTJ* DAØTT
252. P0166 1CF9         JMP* (ENXMT)
253. P0167 0000   IINXM  NUM  0 LØØP TØ INHIBIT XMTR INTR.
254. P0168 C8E7          LDA* ØUTST
255. P0169 A03C          AND- ZR+9  CLR IT.
256. P016A 68E5          STA* ØUTST
257. P016B 580F          RTJ* DAØTT
258. P016C 1CFA          JMP* (IINXM)  RETURN
259. P016D 0000   ØUTP   NUM  0  MUST BE ENTERED WITHE DATA IN Q-REG.
260. P016E C8E1           LDA* ØUTST   DATA ØNLY , IN BITS 0-7 ØF Q.
261. P016F A01B           AND- $1B =$FE00 TØ CLR ØLD DATA+DATA GØ BIT.
262. P0170 0874           EAQ  A  CØMBIND DATA & STATUS.
263. P0171 68DE          STA* ØUTST  SAVE
264. P0172 A03B          AND- ZR+8
265. P0173 B02B          EØR- ØNB+8  SET GØ BIT FØR PULSE ØUT-PUT.
266. P0174 5806           RTJ* DAØTT
267. P0175 A03B          AND- ZR+8  CLR GØ BIT.
268. P0176 A03C          AND- ZR+9
269. P0177 B02C          EØR- ØNB+9  ENABLE FØR NEXT INTRPT.
270. P0178 5802          RTJ* DAØTT
271. P0179 1CF3          JMP* (ØUTP)
```

```
272. P017A 0000   DAØTT NUM  0  LØCAL CALL TØ DAØT (SAVE 2 WØRDS)
 273. P017B 5800         RTJ  DAØT
      P017C FEE8
274. P017D 1CFC         JMP* (DAØTT)
275. P017E C88E   XMTR  LDA* PTR
276. P017F 0822         TRA  Q
277. P0180 09CE         INA  -2*BUF-1  DØNE?
278. P0181 013D         SAM  MUC-*-1
279. P0182 0114         SAN  STXX-*-1
280. P0183 C800         LDA  STXR
      P0184 FF87
281. P0185 B03F         EØR- $3F  ?DATA MØDE
282. P0186 0108         SAZ  MUC-*-1
283. P0187 0C00   STXX  ENQ  0  DØNE SØ CLR STXR.
284. P0188 4800         STQ  STXR
      P0189 FF82
285. P018A 5800         RTJ  CLRS  CLR BIT 0
      P018B 01E1
286. P018C 58DA         RTJ* IINXM
287. P018D 58C3         RTJ* ENIR ENABLE RECVR.
288. P018E 14EA         JMP- (D)
289. P018F EA00   MUC    LDQ  TAB,Q PUT DAT IN Q FØR ØUT-PUT.
       P0190 FF7E
290. P0191 D800         RAØ  PTR
      P0192 FF7A
291. P0193 58D9         RTJ* ØUTP  SEND DATA & ENABLE XMTR FØR NEXT INTE
 292. P0194 14EA         JMP- (D)
293. P0195 FFCD   RLCTR NUM  -50
295. P0196 58C3   FETCH RTJ* IIHR  INHIBIT RECVR. TIL READ.
 296. P0197 5488         RTJ- (ARE)  READ IN DATA NØW.
297. P0198 0045         ADC  $45  WØRD 9
298. P0199 0F46         ARS  6  TØ RT. JSTFY. BITS 6-D.
299. P019A A009         AND- 9  =$7F  REMØVE PARITY FRØM INCØMING
 300. P019B 0822         TRA  Q
301. P019C C8B3         LDA* ØUIST  ADD RESET BIT TØ STATUS.
 302. P019D A03D         AND- ZR+10  SET RESET BIT IN STATUS.
 303. P019E B02D         EØR- ØNB+10
304. P019F 58DA         RTJ* DAØTT  RECVR. ENIR WILL CLEAR IT.
 305. P01A0 0AC3         ENA  -60  WAIT FØR 1 BLK ØF DATA TØ XFR.
 306.                ****THIS IS END ØF PAPER TAPE**********************
 307. P01A1 0000
308. P01A2 6800        STA  TMDØT
      P01A3 00F2
309. P01A4 C800         LDA  LAS
      P01A5 01B0
310. P01A6 09FD         INA  -2  LAST MSG = RU INP
311. P01A7 0101         SAZ  LARUI-*-1
312. P01A8 1835         JMP* CR3
313. P01A9 C800   LARUI LDA  LASCH
      P01AA 00EA
314. P01AB B800         EØR  SP1  LAT CHR=SP1?
      P01AC 008E
315. P01AD 0111         SAN  S1NØT-*-1
316. P01AE 181B         JMP* S1
317. P01AF 0814   S1NØT TRQ  A
318. P01B0 B800         EØR  SP1 IS THIS =SP1?
      P01B1 0089
319. P01B2 0111         SAN  NSSP1-*-1
```

```
320. P01B3 1813           JMP* ISSP1
321. P01B4 C800    NSSP1  LDA  LASCH
     P01B5 00DF
322. P01B6 0FC8           ALS  8
323. P01B7 0874           EAQ  A
324. P01B8 B000           EØR  =N$3F48  LAST 2 CHRS=?HALT..ETC
     P01B9 3F48
325. P01BA 011B           SAN  ISSP1-*-1  IT MAY BE NØISE, IGNØRE.
326. P01BB E02B           LDQ- $2B =256
327. P01BC 5800           RTJ  DEL
     P01BD FE9C
328. P01BE D807    RETRY  RAØ* THIRE
329. P01BF C806           LDA* THIRE
330. P01C0 0122           SAP  KEEP-*-1
331. P01C1 1800           JMP  LA10
     P01C2 0148
332. P01C3 0A06    KEEP   ENA  6  ERR MSG --6
333. P01C4 1815           JMP* MSØUT
334.                      *
335. P01C5 FFFB    THIRE  NUM  -4
336. P01C6 4800    ISSP1  STQ  LASCH
     P01C7 00CD
337. P01C8 1855           JMP* NUS
338. P01C9 0814    S1     TRQ  A  IS THIS SPC #2?
339. P01CA B871           EØR* SP2
340. P01CB 0114           SAN  CR2-*-1  NØ
341. P01CC 5800    GØSTN  RTJ  NULTA
     P01CD 00DA
342. P01CE 1800           JMP  STRIN  YES, GØ CALL ANØTHER STRING.
     P01CF FEE2
343. P01D0 C8C4    CR2    LDA* RLCTR
344. P01D1 0126           SAP  ERT-*-1
345. P01D2 C800           LDA  LSB  YES, GØ REDØ LAST STRING.
     P01D3 FF13
346. P01D4 09E7           INA  -BUF
347. P01D5 6800           STA  LSB
     P01D6 FF10
348. P01D7 18F4           JMP* GØSTN
349. P01D8 0A24    ERT    ENA  $24
350. P01D9 5800    MSØUT  RTJ  MSGG
     P01DA 014A
351. P01DB 1800    ØFFG   JMP  LØGØF
     P01DC 01FA
352. P01DD 0814    CR3    TRQ  A    Q IS SAVED
353. P01DE B009           EØR- 9  =7F IS IT A NULL?
354. P01DF 0103           SAZ  NS-*-1 YES, SØ IGNØRE.
355. P01E0 0814           TRQ  A
356. P01E1 B046           EØR- $46  =$0A  IS IT A LINE FEED.?
357. P01E2 0111           SAN  ØVS1-*-1  IF SØ, IGNØRE IT.
358. P01E3 183A    NS     JMP* NUS
359. P01E4 0814    ØVS1   TRQ  A
360. P01E5 B800           EØR  CR IS IT A C.R.?
     P01E6 00A4
361. P01E7 011C           SAN  ØVR-*-1  NØ,SØ GØ SAVE  THE CHR.
362. P01E8 C800           LDA  PTRP  IS 1ST CHR A C.R.?
     P01E9 00AA
363. P01EA 0106           SAZ  NSS-*-1  YES,IGNØRE LEADING C.R.
```

```
364. P01EB C800          LDA  LASCH
     P01EC 00A8
365. P01ED B800          EØR  CR  WAS LAST CHR SVD A C.R.
     P01EE 009C
366. P01EF 0118          SAN  UNG-*-1  NØ, SØ GØ ØN.
367. P01F0 182D          JMP* NUS
368. P01F1 4800    NSS   STQ  LASCH
     P01F2 00A2
369. P01F3 182A          JMP* NUS
370. P01F4 C800    ØVR   LDA  LASCH
     P01F5 009F
371. P01F6 011D          SAN  NØTIS-*-1
372. P01F7 1826          JMP* NUS
373. P01F8 D800    UNG   RAØ  CRC  BUMP C,R. CTR
     P01F9 0092
374. P01FA C800          LDA  CRC
     P01FB 0090
375. P01FC 60FF          STA- I
376. P01FD 09F7          INA  -8  MØRE THAN 7 C.R.'S.
377. P01FE 0131          SAM  GØØD-*-1  NØ.
378. P01FF 1815          JMP* TRB  YES, GIVE BUF. ØVF MSG.
379. P0200 C800    GØØD  LDA  PTRP  STRING PTR.
     P0201 0092
380. P0202 6900          STA  CRC,I
     P0203 0088
381. P0204 4800    NØTIS STQ  LASCH SAVE LAST CHR
     P0205 008F
382. P0206 0814          TRQ  A
383. P0207 D800          RAØ  PTRP  PACK ALL ØTHERS INTØ IBUF.
     P0208 008B
384. P0209 E800          LDQ  PTRP
     P020A 0089
385. P020B 0FAF          QLS  15  ØDD ØR EVEN.
386. P020C 0161           SQP  EVEN-*-1
387. P020D 181B          JMP* ØDD
388. P020E BA00    EVEN  EØR  IBUF-1,Q  (Q=PTRT/2)
     P020F 038B
389. P0210 6A00          STA  IBUF-1,Q  CØMBIND WITH 1 ST HALF.
     P0211 0389
390. P0212 0D9C    TES   INQ  -INBUF+1  IBUF FULL?
391. P0213 0179          SQM  NUS-*-1  NØ.
392. P0214 D80E    TRB   RAØ* ØULP
393. P0215 C80D          LDA* ØULP
394. P0216 011C          SAN  BRT-*-1  ↑Ø LAST MSG?
395. P0217 C800           LDA  LAS
     P0218 013D
396. P0219 09F5           INA  -10
397. P021A 010B          SAZ  TRYØ-*-1
398. P021B 1800    SENDØ JMP  SECRØ SEND ØUT ↑Ø.
     P021C 01FF
399. P021D 0AF5    NUS   ENA  -10  CLK RESET FØR 10 SEC.
400. P021E 6877          STA* TMDØT
401. P021F 5800          RTJ  ENIR  ENABLE RECVRS NØW.
     P0220 FF30
402. P0221 14EA          JMP- (D)
403. P0222 FFFE    ØULP  NUM  -1
404. P0223 0A03    BRT   ENA  3  BUF ØVF MSG.
```

```
405. P0224 5800          RTJ  MSGG    MSGG ----3
     P0225 00FF
406. P0226 1800    TRYØ  JMP  NREL
     P0227 014E
407. P0228 E86B    ØDD   LDQ* PTRP
408. P0229 0F21          QRS  1  DVI BY 2
409. P022A 0FC8          ALS  8  LEFT JSTFY.
410. P022B 6A00          STA  IBUF,Q  SAVE 1 ST HALF.
     P022C 036F
411. P022D 18E4          JMP* TES
412.               NR3   TIMER TDC-NR3-1,4,X,2
412. P022E 54F4
412. P022F 1124
412. P0230 0004
413. P0231 0000          NUM  0  CALL TDC IN 1 SEC. TØ CHECK TIME-ØUT.
414. P0232 14EA   DIS    JMP- (D)
415. P0233 C863    TDC   LDA* STPC  STØP CLK. REQUESTED?
416. P0234 0131          SAM  NØTN-*-1  NØ.
417. P0235 14EA          JMP- (D)
418. P0236 C85F    NØTN  LDA* TMDØT  TIMED ØUT?
419. P0237 0124          SAP  YETØ-*-1
420. P0238 D85D          RAØ* TMDØT NØ, GØ AGAIN.
421. P0239 18F4          JMP* NR3
422. P023A 002A    SP1   NUM  $2A  SPECIAL CHR FØR PDP 10  TØ TELL
423. P023B 0019    SP2   NUM  $19  US THAT MØRE STRINGS CAN BE
424.                **********
425. P023C 5800    YETØ  RTJ  IIHR  INHIBIT RECIVER INTERRUPTS.
     P023D FF1C
426. P023E 1800          JMP  DEBUG2  REMØVE AFTER DEBUGGING.
     P023F 03D1
427. P0240 0AC3    GG2   ENA  -60
428. P0241 6854          STA* TMDØT  RESET TIMER.
429. P0242 5488          RTJ- (ARE)  SEE IF TELEPHØNE
430. P0243 0045          ADC  $45  LINE HAS BEEN LØST.
431. P0244 A028          AND- ØNB+5
432. P0245 0111          SAN  NT4-*-1
433. P0246 1823          JMP* T4
434. P0247 C06E    NT4   LDA- CØRSTA  CK LØG-IN BIT
435. P0248 0131          SAM  AMLØG-*-1
436. P0249 181E          JMP* NØLØGN
437. P024A C800    AMLØG LDA  LAS
     P024B 010A
438. P024C 09FE          INA  -1  LAST MSG =K/F?
439. P024D 0102          SAZ  NKSF-*-1
440. P024E 09F3          INA  -12  LAST =F?
441. P024F 011C          SAN  NØKSF-*-1
442. P0250 C800    NKSF  LDA  IBUF
     P0251 034A
443. P0252 B000          EØR  =N$2E54  .TØTAL ECT.?
     P0253 2E54
444. P0254 0105          SAZ  JB-*-1
445. P0255 C800          LDA  IBUF
     P0256 0345
446. P0257 B000          EØR  =N$2E4A .JØB XX  ETC.?
     P0258 2E4A
447. P0259 0112          SAN  NØKSF-*-1
448. P025A 1800    JB    JMP  ITBØJ
     P025B 0293
```

```
449. P025C D800   NØKSF  RAØ  HI2  DISCØNNECT - CTR.
     P025D FDE7
450. P025E 0C0F         ENQ  15  CLR BIT.
451. P025F 5800         RTJ  CLRS
     P0260 010C
452. P0261 0C05         ENQ  5 CLR LØGIN REQUEST.P
453. P0262 5800         RTJ  CLRS
     P0263 0109
454. P0264 0C0A         ENQ  10  SET ATTACHING REQ.
455. P0265 5800         RTJ  SETST
     P0266 00FD
456. P0267 1800   NØLØGN JMP  DISC  GØ RE- DIAL.
     P0268 FE26
457.               T4    SCHDLE NR3-T4-1,5,X
457. P0269 54F4
457. P026A 1305
457. P026B 7FC3
458. P026C E81F         LDQ* CRC  NØ ØF C.R.'S
459. P026D C826         LDA* PTRP  CHR PTR.
460. P026E 0111         SAN  NLIN-*-1
461. P026F 1040         JMP* Z  NØT ANY CHRS RECVD.
462. P0270 C800   NLIN  LDA  LAS
     P0271 00E4
463. P0272 09F5         INA  -10  LAST MSG =↑Ø?
464. P0273 0112          SAN  CKCLØ-*-1
465. P0274 1800         JMP  LA10
     P0275 0095
466. P0276 C800   CKCLØ LDA  LAS
     P0277 00DE
467. P0278 09F4         INA  -11  LAST MSG =↑C↑C?
468. P0279 0118         SAN  NØTCLS-*-1
469. P027A C06E          LDA- CØRSTA
470. P027B A02A         AND- ØNB+7
471. P027C 0102         SAZ  SETTY-*-1
472. P027D 1800         JMP  B7  PRØABLY ↑C AFTER ATT,GØE ØUT.
      P027E 00A1
473. P027F 0C0C   SETTY ENQ  12  SEND R SETTTY
474. P0280 1800         JMP  MM00
     P0281 028E
475. P0282 C812   NØTCLS LDA* LASCH  CK IF LAST IS C.R.
476. P0283 B807         EØR* CR
477. P0284 0102         SAZ  YET2-*-1  YES, SØ # ØF LINES=CTR.
 478. P0285 0D01         INQ  1  NØ, SØ BUMP BY ØNE.
479. P0286 0FA1         QLS  1  FØR TWØ WØRD JMP TABLE.
480. P0287 C800   YET2  LDA  IBUF
     P0288 0313
481. P0289 1A0F         JMP* LINN,Q
482. P028A 000D   CR    NUM  $0D  ASCII =C.R.
483. P028B 0000   CRC   NUM  0,0,0,0,0,0,0,0  CRC(0)=# ØF C.R.'S
      P028C 0000
     P028D 0000
     P028E 0000
     P028F 0000
     P0290 0000
     P0291 0000
     P0292 0000
484. P0293 0000   PTRP  NUM  0 CHAR. PTR. IN IBUF TABLE.
485. P0294 0000   LASCH NUM  0  LAST CHR. STØRED.
```

```
486. P0295 FFE6   TMDØT NUM  -25  TIME ØUT CTR.
487. P0296 FFFE   STPC  NUM  -1  STØP CLK FLG - STØP IF +
488. P0297 1300   LINN  JMP  Z  *  NØTE ALL JMPS MUST BE 2
     P0298 0017
489. P0299 1800         JMP  ØNE  *  WØRD INSTRUCTIØNS FØR ADD.
      P029A 0025
490. P029B 1800         JMP  TWØ
     P029C 00F6
491. P029D 1800         JMP  THREE
     P029E 008A
492. P029F 1800         JMP  FØUR
     P02A0 0245
493. P02A1 1800         JMP  FIVE
     P02A2 0209
494. P02A3 1800         JMP  SIX
     P02A4 0211
495. P02A5 1800         JMP  SEVEN
     P02A6 0210
496. P02A7 0000   NULTA NUM  0  CRC(0) TØ LASCH ( TØTAL ØF $A CELLS)
 497. P02A8 0A00         ENA  0
498. P02A9 0C09         ENQ  9
499. P02AA 6AE0   NUL2  STA* CRC,Q  CLK START FLG LEFT SET.
 500. P02AB 0DFE         INQ  -1
501. P02AC 0171         SQM  BA2-*-1  ALSØ CLK STØP IS ØN.
502. P02AD 18FC         JMP* NUL2  BUT NUS CAN TURN ØN CLK
503. P02AE 1CF8   BA2   JMP* (NULTA)  IF IT IS REQUESTED.
504. P02AF C800   Z     LDA  LAS  # ØF LAST MSG SENT TØ ID
     P02B0 00A5
505. P02B1 0113         SAN  LØGI-*-1
506. P02B2 1800         JMP  NØT  YES, TRY AGAIN.
     P02B3 FDD4
507. P02B4 FFFC   LØØ   NUM  -3
508. P02B5 C8FE   LØGI  LDA* LØØ
509. P02B6 0124         SAP  ERNS-*-1
510. P02B7 D8FC         RAØ* LØØ
511. P02B8 5800         RTJ  ENIR
     P02B9 FE97
512. P02BA 14EA         JMP- (D)
513. P02BB 0A23   ERNS  ENA  $23  MSG-3
514. P02BC 5868         RTJ* MSGG
515. P02BD 1800         JMP  TRYØFF
     P02BE 0116
516. P02BF 0822   ØNE   TRA  Q
517. P02C0 C8D3         LDA* LASCH
518. P02C1 09D1         INA  -$2E  ? LAST CHR =.
519. P02C2 0117         SAN  NØTNP-*-1
520. P02C3 C800         LDA  LAS  LAST= ATTACH?
     P02C4 0091
521. P02C5 09F7         INA  -8
522. P02C6 0112         SAN  GØNP-*-1
523. P02C7 1800         JMP ERB PRØBABLY=?TTY ALREADY ATTACHED.
      P02C8 0182
524. P02C9 1817   GØNP  JMP* NP
525. P02CA 0814   NØTNP  TRQ  A
526. P02CB B000         EØR  =N$5061  PASSWØRD?
     P02CC 5061
527. P02CD 010C         SAZ   ITPAS-*-1  YES,
528. P02CE 0814         TRQ  A  1 ST  CHR=CØNFIRM:
```

```
529. P02CF B000          EØR   =N$434F
     P02D0 434F
530. P02D1 0117          SAN   JMEM-*-1
531. P02D2 C800          LDA   LAS
     P02D3 0082
532. P02D4 09FE          INA   -1  LAST MSG =K/F?
533. P02D5 0113          SAN   JMEM-*-1
534. P02D6 0C0D          ENQ   13  SEND MSG= F
535. P02D7 1800          JMP   MMØØ
     P02D8 0237
536. P02D9 1848    JMEM  JMP*  EM
537. P02DA 0C02    ITPAS ENQ   2 SET STATUS =SENDING PASSWØRD.
538. P02DB 5800           RTJ   SETST
     P02DC 0087
539. P02DD 0C04          ENQ   4     SEND MSG 4 = PASS
540. P02DE 1800          JMP   MMØØ
     P02DF 0230
541. P02E0 C875    NP    LDA*  LAS
542. P02E1 09FB    N10   INA   -4  WAS IT 4(RCKRCG)?
543. P02E2 0111          SAN   NØT12-*-1
544. P02E3 180D          JMP*  LA4
545. P02E4 0901    NØT12 INA   1 DØNE MSG?
546. P02E5 0112          SAN   NNLA3-*-1
547. P02E6 1800          JMP   RETRY
     P02E7 FED6
548. P02E8 0902    NNLA3 INA   2  WAS IT 1?
549. P02E9 0112          SAN   ERRMØ-*-1
550. P02EA 1800          JMP   LØGØF  GØ SEND K/F
     P02EB 00EB
551. P02EC 0A1F    ERRMØ ENA   $1F  MSGG ---Ø
552. P02ED 5837          RTJ*  MSGG   ---Ø
553. P02EE 1865          JMP*  TØF2  (TRYØFF)
554. P02EF 181B    JPL10 JMP*  LA10
555. P02F0 0C0A    LA4   ENQ   10
556. P02F1 587B          RTJ*  CLRS  CLEAR 10
557. P02F2 0C06          ENQ   6  SIGNED-ØFF BIT.
558. P02F3 5879          RTJ*  CLRS
559. P02F4 0C0F          ENQ   15
560. P02F5 586E          RTJ*  SETST  LØG-IN. BIT.
561. P02F6 0C02          ENQ   2  CLR - PASSWØRD BIT.
562. P02F7 5875          RTJ*  CLRS
563. P02F8 0C0B          ENQ   11  SEND CØNTRØL C.
564. P02F9 1868          JMP*  MMØØI
565. P02FA 0C09    LA3   ENQ   9  ALL DØNE
566. P02FB 5871          RTJ*  CLRS
567. P02FC 0C07          ENQ   7  SET SIGNING ØFF
568. P02FD 5866          RTJ*  SETST
569. P02FE C000          LDA   =N$4800
     P02FF 4800
570. P0300 6400          STA   FRY  RESET LSB
     P0301 7F41
571.               DMSG  FWRITE $E,B7-DMSG-1,M2-DMSG-1,11,A,4,4,,X
571. P0302 54F4
571. P0303 0D44
571. P0304 001C
     P0305 0000
571. P0306 100E  1004
571. P0307 000B
     P0308 028A
```

```
572. P0309 14EA          JMP- (D)
573. P030A C06E   LA10   LDA- CØRSTA
574. P030B A02A          AND- ØNB+7
575. P030C 0101          SAZ  NB76-*-1
576. P030D 1812          JMP* B7
577. P030E 0C0D   NB7S   ENQ  13  RUN INPUT.
578. P030F 5854          RTJ* SETST
579. P0310 C400          LDA  FRY
     P0311 7F41
580. P0312 9000          SUB  =N$4800
     P0313 4800
581. P0314 0C00          ENQ  0
582. P0315 3000          DVI  =N$30
     P0316 0030
583. P0317 6800          STA  ME3+2 IMBEDD IN DØNE MSG.
     P0318 023C
584. P0319 C000          LDA  =N$4800
     P031A 4800
585. P031B 6800          STA  LSB
     P031C FDCA
586. P031D 0C02          ENQ  2  RU INP MSG
587. P031E 1843          JMP* MMØØI
588. P031F 0C01   B7      ENQ  1
589. P0320 1841          JMP* MMØØI  K/F MSG.
590. P0321 0A18   EM      ENA  $18  MSGG H
591. P0322 5802          RTJ* MSGG   - H
592. P0323 1830          JMP* TØF2
593. P0324 0000   MSGG   NUM  0
594. P0325 5800          RTJ  MSGGI
     P0326 01A0
595. P0327 1CFC          JMP* (MSGG)
596. P0328 0822   THREE  TRA  Q
597. P0329 C82C          LDA* LAS
598. P032A 09F6          INA  -9  LAST MSG=9(SYS)
599. P032B 0112          SAN  NL9-*-1
600. P032C 1800          JMP  IS9
     P032D 015C
601. P032E 0814   NL9    TRQ  A
602. P032F B000          EØR  =N$4941  IACC ... ETC.
     P0330 ~~4941~~ 6961
603. P0331 0111          SAN  NIAC-*-1
604. P0332 1811          JMP* IACC
605. P0333 0814   NIAC   TRQ  A
606. P0334 B000          EØR  =N$5361  SAVED ALL ... ETC.
     P0335 5361
607. P0336 0104          SAZ  TIZ3-*-1
608. P0337 0814          TRQ  A
609. P0338 B000          EØR  =N$2E4A   .JØB XX  ETC.
     P0339 2E4A
610. P033A 0115          SAN  NØT3-*-1
611. P033B C81A   TIZ3   LDA* LAS
612. P033C 09FE          INA  -1  LAST MSG = K/F?
613. P033D 0103          SAZ  LAS13-*-1
614. P033E 09F3          INA  -12  WAS IT MSG 13 "F"?
615. P033F 0101          SAZ  LAS13-*-1
616. P0340 181E   NØT3   JMP* ERM
617. P0341 1800   LAS13  JMP  ITBØJ
     P0342 01AC
```

```
618. P0343 0C03   IACC  ENQ  3  YES, CLR BIT 3
619. P0344 5828         RTJ* CLRS
620. P0345 C06E   CKRE  LDA- CØRSTA  IS BIT 10 =ATTACH REQ?
621. P0346 A02D           AND- ØNB+10
622. P0347 011E         SAN  T10-*-1
623. P0348 C06E         LDA- CØRSTA
624. P0349 A028         AND- ØNB+5
625. P034A 0101         SAZ  NIST5-*-1
626. P034B 1811         JMP* IST5
627. P034C C06E   NIST5 LDA- CØRSTA  IS IT STATUS REQ.?
628. P034D A024         AND- ØNB+1
629. P034E 0102         SAZ  ERML-*-1
630. P034F 0C09         ENQ  9  SEND STAT. MSG.
631. P0350 1811         JMP* MMØØI
632. P0351 0A1C   ERML  ENA  $1C  MSG  -L
633. P0352 58D1         RTJ* MSGG  --L
634. P0353 1800   TØF2  JMP  TRYØFF
     P0354 0080
635. P0355 0000   LAS   NUM  0
636. P0356 C400   T10   LDA  SYSTAT+1  JØB #
     P0357 7F4F
637. P0358 6800         STA  ME8+2
     P0359 0207
638. P035A 0C08         ENQ  8
639. P035B 1806         JMP* MMØØI  TØ ATTACH-MSG.
640. P035C 0C07   IST5  ENQ  7  LØGIN MSG.
641. P035D 1804         JMP* MMØØI
642. P035E 0A1D   ERM   ENA  $1D
643. P035F 58C4         RTJ* MSGG  --M
644. P0360 1874         JMP* TRYØFF
645. P0361 1800   MMØØI JMP  MMØØ
     P0362 01AD
646. P0363 0B00   SETST  NØP  0  TØ SET BIT NØ. IN Q REG.
647. P0364 C400         LDA  SYSTAT
     P0365 7F4E
648. P0366 A233         AND- ZR,Q
649. P0367 B223         EØR- ØNB,Q
650. P0368 6400         STA  SYSTAT
     P0369 7F4E
651. P036A 606E         STA- CØRSTA
652. P036B 1CF7         JMP* (SETST)
653. P036C 0B00   CLRS  NØP  0  TØ CLR STATUS BIT SET IN Q REG.
654. P036D C400          LDA  SYSTAT
     P036E 7F4E
655. P036F A233         AND- ZR,Q
656. P0370 6400         STA  SYSTAT
     P0371 7F4E
657. P0372 606E         STA- CØRSTA
658. P0373 1CF8         JMP* (CLRS)
659. P0374 FFFD   LØPC  NUM  -2  CTR USED BY N8
660. P0375 0A00   NREL  ENA  0  DIS-CØNNECT.
661. P0376 5800         RTJ  DAØT  DISCØNNECT MØDEM
     P0377 FCED
662. P0378 C06E         LDA- CØRSTA  SET EXIT STATUS TØ SAVE ØNLY
663. P0379 A000          AND  =N$84E2  SAVE 15,10,7,5,6,1
     P037A 84E2
664. P037B 606E         STA- CØRSTA
665. P037C 6400         STA  SYSTAT
     P037D 7F4E
```

```
666. P037E 0A00          ENA  0
667. P037F 6800          STA  STPC  STØP CLK
     P0380 FF15
668. P0381 0C78          ENQ  120  2 SEC PAUSE.
669. P0382 5800          RTJ  DEL  PAUSE.
     P0383 FCD6
670. P0384 C400          NUM  $C400,$137,$6400,$13B,$6400,$13F   RESTØRE
      P0385 0137
     P0386 6400
     P0387 013B
     P0388 6400
     P0389 013F
671. P038A E807          LDQ* DIALAD
672. P038B 0D01          INQ  1
673. P038C 0A00          ENA  0
674. P038D 66EB          STA- (SYSLIB),Q  NULL LIB ENTRY.
675.              RELL   RELEAS (START-RELL-1),T,X
675. P038E 54F4
675. P038F 1901
675. P0390 FC70
676. P0391 7FFF X DIALAD ADC  DIALUP
677. P0392 B000   TWØ    EØR  =N$3F43  ?CAN'T ATTACH
     P0393 3F43
678. P0394 011D          SAN   NQ-*-1  NØ
679. P0395 C8BF          LDA* LAS
680. P0396 09F7          INA  -8  WAS IT =8
681. P0397 0111          SAN  N8-*-1
682. P0398 1812          JMP* IN8  YES
683. P0399 E8BB   N8     LDQ* LAS
684. P039A C8D9          LDA* LØPC
685. P039B 0102          SAZ  ERJ-*-1
686. P039C D8D7          RAØ* LØPC
687. P039D 18C3          JMP* MMØØI
688. P039E 0A1A   ERJ    ENA  $1A  MSG=J
689. P039F 5884          RTJ* MSGG
690. P03A0 1800          JMP  TRYØFF
     P03A1 0033
691. P03A2 C8B2   NQ     LDA* LAS
692. P03A3 09F8          INA  -7  =7?
693. P03A4 0111          SAN  NY7-*-1
694. P03A5 1813          JMP* Y7
695. P03A6 09FE   NY7    INA  -1  ?8
696. P03A7 0102          SAZ  IN8-*-1
697. P03A8 1800          JMP RN8
     P03A9 00B4
698. P03AA 0C0A   IN8    ENQ  10
699. P03AB 58C0          RTJ* CLRS  CLR ATTACHING BIT.
700. P03AC C800          LDA  IBUF
     P03AD 01EE
701. P03AE B000          EØR  =N$3F43  ?CAN'T ATTACH
     P03AF 3F43
702. P03B0 0101          SAZ  ANT-*-1
703. P03B1 186D          JMP* CKTT NØT=?C
704. P03B2 0A19   ANT    ENA  $19
705. P03B3 5800          RTJ  MSGG   --I
     P03B4 FF6F
706. P03B5 1800          JMP  S  TRY TAKING STATUS.
     P03B6 009A
```

```
707. P03B7 0000   TTJØ  NUM  0
708. P03B8 0C05   Y7    ENQ  5  CLR LØGIN REQ.
709. P03B9 58B2        RTJ* CLRS
710. P03BA C800        LDA  IBUF  1 ST CHRS =
     P03BB 01E0
711. P03BC B000        EØR  =N$4A4F  JØB---
     P03BD 4A4F
712. P03BE 0102        SAZ  FA4F
713. P03BF 1800        JMP  KJT  NØT = JØ
     P03C0 0082
714. P03C1 C800   FA4F  LDA  IBUF+2  GET JØB #
     P03C2 01DB
715. P03C3 68F3       STA* TTJØ  TEMP SAVE
716.              SCH   SCHDLE ITPAS-SCH-1,5,X  GØ SEND PASSWØRD.
 716. P03C4 54F4
716. P03C5 1305
716. P03C6 7F14
717.              T9    TIMER NUP-T9-1,5,X,2
717. P03C7 54F4
717. P03C8 1125
717. P03C9 0005
718. P03CA 0003        NUM  3  4 SEC.
719. P03CB 14EA        JMP- (D)
720. P03CC FFFA   LØP7  NUM  -5  USED BY NUP
721. P03CD C8FE   NUP   LDA* LØP7   3 TIMES.
722. P03CE 0102        SAZ  ERF-*-1  YES, ERRØR.
723. P03CF D8FC        RAØ* LØP7
724. P03D0 1816        JMP* ITS4
725. P03D1 0A16   ERF   ENA  $16  MSG- F
726. P03D2 5800        RTJ  MSGGI     -F
     P03D3 00F3
727. P03D4 C06E   TRYØFF LDA- CØRSTA  LØGGED IN?
728. P03D5 012B        SAP  ITSREL-*-1
729. P03D6 0A00   LØGØF ENA  0
730. P03D7 6800        STA  STXR  CLR BUSY FLG.
     P03D8 FD33
731. P03D9 0C07        ENQ  7
732. P03DA 5800        RTJ  SETST
     P03DB FF87
733. P03DC C809        LDA* REZ2
734. P03DD 0134        SAM  MØRE-*-1
735. P03DE 0A29        ENA  $29
736. P03DF 5800        RTJ  MSGGI  MSG - - Z Z Z
     P03E0 00E6
737. P03E1 1893   ITSREL JMP* NREL
738. P03E2 D803   MØRE  RAØ* REZ2
739. P03E3 1800        JMP  B7  GØ SIGN ØFF.
     P03E4 FF3A
740.              *****
741. P03E5 FFFE   REZ2  NUM  -1
742. P03E6 C800   ITS4  LDA  PTRP  > 10 CHRS RECVD. YET?
     P03E7 FEAB
743. P03E8 09F5        INA  -10
744. P03E9 0121         SAP  P10-*-1
745. P03EA 18DC        JMP* T9
746. P03EB 0C02   P10   ENQ  2  CLR PASSWØRD
747. P03EC 5800        RTJ  CLRS
     P03ED FF7E
```

```
748. P03EE 0AFC          ENA  -3
749. P03EF 68DC          STA* L0P7
750. P03F0 C800          LDA  IBUF
     P03F1 01AA
751. P03F2 A835          AND* F0F0
752. P03F3 B835          E0R* T0T0
753. P03F4 0111          SAN  BUFI-*-1
754. P03F5 1810          JMP* THN0
755. P03F6 C800   BUFI   LDA  IBUF
     P03F7 01A4
756. P03F8 B000          E0R  =N$4F74  0THER J0B ...ETC.
     P03F9 4F74
757. P03FA 0116          SAN  ERMG-*-1
758. P03FB 5800          RTJ  IIHR
     P03FC FD5D
759. P03FD E02C          LDQ- $2C
760. P03FE 5800          RTJ  DEL  PAUSE 8.5 SEC.
     P03FF FC5A
761. P0400 1850          JMP* S  G0 D0 STATUSING.
762. P0401 0A17   ERMG   ENA  $17   MSG---G
763. P0402 5800         RTJ  MSGGI
     P0403 00C3
764. P0404 18CF          JMP* TRY0FF
765. P0405 5800   THN0   RTJ  IIHR  INHIBIT RECVR.
     P0406 FD53
766. P0407 C8AF          LDA* TTJ0
767. P0408 6400          STA  SYSTAT+1  J0B #
     P0409 7F4F
768. P040A C800          LDA  IBUF  SAVE TIME 0F DAT F0R D0NE MSG.
     P040B 0190
769. P040C 6800          STA  M2+9
     P040D 0189
770. P040E C800          LDA  IBUF+1
     P040F 018D
771. P0410 6800           STA  M2+10
     P0411 0186
772. P0412 0C0F          ENQ  15
773. P0413 5800          RTJ  SETST  L0GGED IN
     P0414 FF4E
774. P0415 0C06          ENQ  6  CLR SGNED-0FF.
775. P0416 5800          RTJ  CLRS
     P0417 FF54
776. P0418 0C02          ENQ  2
777. P0419 5800          RTJ  CLRS
     P041A FF51
778. P041B 0C0A   SECR0  ENQ  10  SEND C0NTR0L -0 MSG T0 10
779. P041C 1800          JMP  MM00
     P041D 00F2
780. P041E C800   CKTT   LDA  IBUF+1  /TTY ALREADY..ETC
     P041F 017D
781. P0420 B000          E0R  =N$5459
     P0421 5459
782. P0422 0116          SAN  ERK-*-1  N0T=
783. P0423 0A13          ENA  $13
784. P0424 5800          RTJ  MSGG   MSG--- C
     P0425 FEFE
785. P0426 1858          JMP* TB0J
786. P0427 F0F0   F0F0   NUM  $F0F0
787. P0428 3030   T0T0   NUM  $3030
```

```
788. P0429 C800   ERK   LDA  IBUF
     P042A 0171
789. P042B B000         EØR  =N$3F49  ?ILLIGAL JØB --
     P042C 3F49
790. P042D 0111         SAN  NRKK-*-1
791. P042E 1822         JMP* S
792. P042F C800   NRKK  LDA  IBUF  1 ST CHR=?N
     P0430 016B
793. P0431 B000         EØR  =N$3F4E
     P0432 3F4E
794. P0433 0105         SAZ  INEK-*-1
795. P0434 C800         LDA  IBUF+1
     P0435 0167
796. P0436 B000         EØR  =N$6F6F  ?TØØ FEW ARG...ETC.
     P0437 6F6F
797. P0438 0115          SAN  ERKK-*-1
798. P0439 0C0A   INEK  ENQ  10  CLR ATTACH BIT&GØ LØGIN
799. P043A 5800         RTJ  CLRS
     P043B FF30
800. P043C 1800         JMP  IST5
     P043D FF1E
801. P043E 0A1B   ERKK  ENA  $1B   MSG  -K
802. P043F 5800         RTJ  MSGGI  MSG--  -K
     P0440 0086
803. P0441 1892         JMP* TRYØFF
804. P0442 0C05   KJT   ENQ  5  CLR  LØGGING BIT.
805. P0443 5800         RTJ  CLRS
     P0444 FF27
806. P0445 C800         LDA  IBUF+4
     P0446 0159
807. P0447 B000         EØR  =N$4B4A  ?=KJ  (KJØB)
     P0448 4B4A
808. P0449 0114         SAN  ERD-*-1
809. P044A 0A12   ERB    ENA  $12
810. P044B 587B         RTJ* MSGGI  MSG--- B
811. P044C 1889         JMP* LØGØF
812. P044D FFFC   SLØPC NUM  -3  USED BY S
813. P044E 0A14   ERD    ENA  $14  MSG --D
814. P044F 5877         RTJ* MSGGI    -D
815. P0450 C8FC   S     LDA* SLØPC
816. P0451 0127         SAP  ERXX-*-1
817. P0452 D8FA         RAØ* SLØPC
818. P0453 0C01         ENQ  1  SET STATUSING BIT.
819. P0454 5800         RTJ  SETST
     P0455 FF0D
820. P0456 0C09         ENQ  9  TRY TAKING STATUS.
821. P0457 1800         JMP  MMØØ
     P0458 00B7
822. P0459 0A28   ERXX  ENA  $28
823. P045A 586C         RTJ* MSGGI  MSG X
824. P045B 1800   TRYØ3 JMP  TRYØFF LØØPING IN S.
     P045C FF77
825. P045D C800   RN8.  LDA  LAS
     P045E FEF6
826. P045F 09F6         INA  -9
827. P0460 0111          SAN  ISN9-*-1
828. P0461 1828         JMP* IS9
829. P0462 09FC   ISN9  INA  -3  LAST =12(R SETTTY)
830. P0463 0111         SAN  NØTM10-*-1
```

```
831. P0464 181B          JMP* ITS10
832. P0465 090B    NØTM10 INA  11  LAST =11(K/F)?
833. P0466 010A           SAZ  KSLF-*-1
834. P0467 09FD           INA  -2  LAS=3(DØNE)
835. P0468 0115           SAN  ERQQ-*-1
836. P0469 C800           LDA  IBUF
     P046A 0131
837. P046B B000           EØR  =N$3F48  ?HALT...
     P046C 3F48
838. P046D 0101            SAZ  1
839. P046E 1817    ERQQ   JMP* ERQ
840. P046F 1800           JMP  RETRY
     P0470 FD4D
841. P0471 C800    KSLF   LDA  IBUF
     P0472 0129
842. P0473 B000           EØR  =N$3F4E  ?NØT A JØB..ETC
     P0474 3F4E
843. P0475 0108           SAZ  TBØJ-*-1
844. P0476 C800           LDA  IBUF
     P0477 0124
845. P0478 B000           EØR  =N$434F  CØNFIRM=LAST CH.?
     P0479 434F
846. P047A 011A           SAN  ERQ-*-1
847. P047B 0C0D           ENQ  13  SEND "F"
848. P047C 1800           JMP  MMØØ
     P047D 0092
849. P047E 1370    TBØJ   JMP* ITBØJ
850. P047F C800    ITS10  LDA  LASCH
     P0480 FE13
851. P0481 09D1           INA  -$2E  LAST CHR AN .?
852. P0482 0112           SAN  ERQ-*-1
853. P0483 1800           JMP  LA10  GØ SEND RU INP
     P0484 FE85
854. P0485 0A21    ERQ    ENA  $21
855. P0486 5840           RTJ* MSGGI  MSG -Q
856. P0487 1800           JMP  TRYØFF
     P0488 FF4B
857. P0489 0C01    IS9    ENQ  1 CLR STATUS REQ.
858. P048A 5800           RTJ  CLRS
     P048B FEE0
859. P048C C800            LDA  IBUF
     P048D 010E
860. P048E A898           AND* FØFØ  =$FOFO
861. P048F B398           EØR* TØTØ  =$3030
862. P0490 0119           SAN  NNØ5-*-1  1 ST CHRS NØT NØ'S.
863. P0491 C800           LDA  IBUF
     P0492 0109
864. P0493 6400           STA  SYSTAT+1  SAVE AS JØB #
     P0494 7F4F
865. P0495 0C0A           ENQ  10
866. P0496 5800           RTJ  SETST
     P0497 FECB
867. P0498 1800           JMP  T10  GØ SEND ATTACH MSG.
     P0499 FEBC
868. P049A C800    NNØ5   LDA  IBUF
     P049B 0100
869. P049C B000           EØR  =N$2E4B  .KJØB---?
     P049D 2E4B
870. P049E 0119           SAN  ERE-*-1  NØT =
```

```
871. P049F C06E          LDA- CØRSTA
872. P04A0 A02A          AND- ØNB+7  SIGINING ØFF?
873. P04A1 0101          SAZ  GØBØJ-*-1
874. P04A2 184C          JMP* ITBØJ
875. P04A3 0C05   GØBØJ  ENQ  5
876. P04A4 5800          RTJ  SETST
     P04A5 FEBD
877. P04A6 0C07          ENQ  7
878. P04A7 1868          JMP* MMØØ
879. P04A8 0A15   ERE    ENA  $15
880. P04A9 581D          RTJ* MSGGI  MSG -E
881. P04AA 18B0          JMP* TRYØ3  (TRYØFF)
882. P04AB C800   FIVE   LDA  LAS
     P04AC FEA8
883. P04AD 09FE          INA  -1  LAST MSG=K/F?
884. P04AE 0117          SAN  SEVEN-*-1
885. P04AF C800          LDA  IBUF
     P04B0 00EB
886. P04B1 B000          EØR  =N$2E54  1ST CH=.TØTAL ETC.
     P04B2 2E54
887. P04B3 0116          SAN  ERIR-*-1
888. P04B4 183A          JMP* ITBØJ
889. P04B5 0B00   SIX    NØP  0
890. P04B6 C800   SEVEN  LDA  LAS
     P04B7 FE9D
891. P04B8 09F6          INA  -9
892. P04B9 0103          SAZ  LL10-*-1
893. P04BA 0A22   ERIR   ENA  $22
894. P04BB 580B          RTJ* MSGGI  MSG -  R
895. P04BC 189E          JMP* TRYØ3  (TRYØFF)
896. P04BD C800   LL10   LDA LASCH
     P04BE FDD5
897. P04BF 09D1          INA  -$2E  ? LAST CHR.= .
898. P04C0 0101          SAZ  ITPE-*-1
899. P04C1 18F8          JMP* ERIR
900. P04C2 1800   ITPE   JMP  LA10  TREAT LIKE 2 LINE RESPØNSE.
     P04C3 FE46
901. P04C4 2030   ASCH   NUM  $2030
902. P04C5 FFFA   CTLP   NUM  -5
903. P04C6 0000   MSGGI  NUM  0  BUFFER 5 MSG REPEATS & RETURN.
904. P04C7 88FC          ADD* ASCH  MAKE IT LEGAL ASCII.
905. P04C8 6800          STA  M+12
     P04C9 00BB
906. P04CA 5800          RTJ  IINXM
     P04CB FC9B
907. P04CC 5800          RTJ  IIHR
     P04CD FC8C
908. P04CE C400          LDA  SYSTAT
     P04CF 7F4E
909. P04D0 5489          RTJ- (AHEX)
910. P04D1 80B5          ADC  (M+14-*)
911. P04D2 C06E          LDA- CØRSTA
912. P04D3 5489          RTJ- (AHEX)
913. P04D4 80B5          ADC  (M+17-*)
914.               MSGK  FWRITE $E,MADD-MSGK-1,M-MSGK-1,20,A,5,5,,X
914. P04D5 54F4
914. P04D6 0D55
914. P04D7 0007
     P04D8 0000
```

```
914. P04D9 100E
914. P04DA 0014
     P04DB 00A2
915. P04DC 14EA          JMP- (D)
916. P04DD 0AE6   MADD   ENA  -25
917. P04DE 6800          STA  TMDØT
     P04DF FDB5
913. P04E0 C8E4          LDA* CTLP
919. P04E1 0122          SAP  RE7-*-1  SKIP IF TYPED 5 TIMES.
 920. P04E2 D8E2          RAØ* CTLP
921. P04E3 18F1          JMP* MSGK
922. P04E4 1CE1   RE7     JMP* (MSGGI)
923. P04E5 E800   FØUR   LDQ  LAS
     P04E6 FE6E
924. P04E7 0DFC          INQ  -3  LAST MSG=3?
925. P04E8 015E          SQN  CKIF-*-1
926. P04E9 B000           EØR  =N$4350 (CPU) ETC.?
     P04EA 4350 454E
927. P04EB 0113          SAN  CKIF-*-1
928. P04EC 1800          JMP  LA3
     P04ED FE0C
929. P04EE 0A00   ITBØJ  ENA  0
930. P04EF 6400          STA  SYSTAT+1
     P04F0 7F4F
931. P04F1 C029          LDA- $29  =$0040  NØRMAL STATUS -ØFF
 932. P04F2 606E           STA- CØRSTA
933. P04F3 6400          STA  SYSTAT   RESTØRE BØTH TØ NØRMAL.
      P04F4 7F4E
934. P04F5 1800          JMP  NREL
     P04F6 FE7E
935. P04F7 0DFB   CKIF   INQ  -4  LAST=LØGIN?
936. P04F8 0152          SQN  ERRP-*-1
937. P04F9 1800          JMP  Y7
     P04FA FEBD
938. P04FB 0A20   ERRP   ENA  $20  MSGG --P
939. P04FC 58C9          RTJ* MSGGI
940. P04FD 1800   TØØF   JMP  TRYØFF
     P04FE FED5
941. P04FF 0B00   MEL    NØP  0
942. P0500 0003          ADC   ME2-ME1
943. P0501 0004          ADC  ME3-ME2
944. P0502 0003          ADC  ME4-ME3  LENGTH ØF 3
945. P0503 0003          ADC  ME7-ME4
946. P0504 0000          ADC  ME7-ME7
947. P0505 0000          ADC  ME7-ME7
948. P0506 0006          ADC  ME8-ME7
949. P0507 0009          ADC  ME9-ME8
950. P0508 0007          ADC  ME10-ME9
951. P0509 0001          ADC  ME11-ME10
952. P050A 0001          ADC  ME12-ME11
953. P050B 0004          ADC  ME13-ME12
954. P050C 0002          ADC  MEMA-ME13
955. P050D 0000   THS    NUM  0
956. P050E FFCD   FIVF   NUM  -50
957. P050F 48FD   MMØØ   STQ* THS  ENTERED WITH MSG NØ IN Q.
958. P0510 0ACD   TMØØ   ENA  -50
959. P0511 68FC          STA* FIVF
960. P0512 C800   TM02   LDA  STXR  CHECK BUSY FLG.
     P0513 FBF8
```

```
961. P0514 010B          SAZ  ZNZ1-*-1
962. P0515 0C3B          ENQ  59  FØR 1 SEC.
963. P0516 5800          RTJ  DEL  PAUSE
     P0517 FB42
964. P0518 C8F5          LDA* FIVF
965. P0519 D8F4          RAØ* FIVF
966. P051A 0134          SAM  ZNZ-*-1
967. P051B 0A27          ENA  $27
968. P051C 58A9          RTJ* MSGGI  MSG  - W
969. P051D 1800          JMP  TRYØFF
     P051E FEB5
970. P051F 18F2  ZNZ     JMP* TMØ2
971. P0520 E8EC  ZNZ1    LDQ* THS
972. P0521 4800          STQ  STXR  SET BUSY
     P0522 FBE9
973. P0523 CADB          LDA* MEL,Q  GET MSG.
974. P0524 4800           STQ  LAS
     P0525 FE2F
975. P0526 0FC1          ALS  1
976. P0527 68E5          STA* THS
977. P0528 0A00           ENA  0
978. P0529 0DFE  ADD     INQ  -1
979. P052A 0142          SQZ  FIN-*-1
980. P052B 8AD3          ADD* MEL,Q  CALC. I ST WØRD ADD ØF MESS. BLK.
981. P052C 18FC           JMP* ADD  BY ADDING LENGTH ØF BLØCKS.
982. P052D 60FF   FIN     STA- I I NØW HAS 1 ST CHR ADD, W/R/T ME1.
983. P052E 0A30           ENA  2*BUF
984. P052F 98DD          SUB* THS  NØ. CARS IN THIS MESSAGE.
985. P0530 0822          TRA  Q  USE Q FØR TAB INDEX.
986. P0531 6800          STA  PTR  START XFR AT THIS PTR VAL.
     P0532 FBDA
987. P0533 C918  LP7     LDA* ME1,I  GET 1 ST.
988. P0534 0F48          ARS  8  EXPAND INTØ 8 BIT WØRDS.
989. P0535 A00A          AND- 10  = FF  FØR SENDING TØ PDP-10
990. P0536 6A00          STA  TAB,Q
     P0537 FBD7
991. P0538 C913          LDA* ME1,I  GET ØUHER HALF.
992. P0539 A00A          AND- 10
993. P053A 6A00          STA  TAB+1,Q
     P053B FBD4
994. P053C 0DD0          INQ  -2*BUF+1  DØNE?
995. P053D 0163          SQP  LP9-*-1
996. P053E 0D31          INQ  2*BUF+1  RESTØRE Q +2
997. P053F D0FF          RAØ- I
998. P0540 18F2          JMP* LP7
999. P0541 0A0D  LP9     ENA  $D  ASCII FØR C.R.
000. P0542 6800          STA  TAB+2*BUF  PUTS C.R. AFTER LAST TAB WD.
     P0543 FBFB
001. P0544 5800          RTJ  NULTA  NULL CØUNTERS.
     P0545 FD61
002. P0546 0C00          ENQ  0  STRING  IN USE.
003. P0547 5800          RTJ  SETST
     P0548 FE1A
004. P0549 1800          JMP  DEBUG  REPLACE BY JMP XMTR AFTER DEBUG.
     P054A 00B6
005. P054B 0303  ME1     NUM  $0303  ↑C↑C
006. P054C 4B2F          ALF  2,K/F  SIGN ØFF.
     P054D 4620
```

```
007. P054E 0303    ME2    NUM   $0303  ↑C↑C
008. P054F 5255           ALF   3,RU INP  RUN INPUTTING PRØG.
     P0550 2049
     P0551 4E50
009. P0552 444F    ME3    ALF   3,DØNE    TØ STØP PDP-10.
     P0553 4E45
     P0554 2020
010. P0555 5243    ME4    ALF   3,RCKRCG  =PASSWØRD
     P0556 4B52
     P0557 4347
011. P0558 4C4F    ME7    ALF   6,LØGI 6415,26
     P0559 4749
     P055A 2036
     P055B 3431
     P055C 352C
     P055D 3236
012. P055E 4154    ME8    ALF   4,ATT
     P055F 5420
     P0560 2020
     P0561 2020
013. P0562 5B36           NUM   $5B36,$3431,$352C,$3236,$5D20  =(6515,26)
      P0563 3431
     P0564 352C
     P0565 3236
     P0566 5D20
014. P0567 5359    ME9    ALF   2,SYS
     P0568 5320
015. P0569 5B36           NUM   $5B36,$3431,$352C,$3236,$5D20
     P056A 3431
     P056B 352C
     P056C 3236
     P056D 5D20
016. P056E 0F20    ME10   NUM   $0F20  ↑Ø
017. P056F 0303    ME11    NUM  $0303  ↑C↑C
018. P0570 5220    ME12   ALF   4,R SETTY
     P0571 5345
     P0572 5454
     P0573 5459
019. P0574 4620    ME13   ALF   2,F
     P0575 2020
020. P0576 0B00    MEMA   NØP   0  THIS MUST BE LAST MESG. LINE.
021. P0577 0000            NUM   0
022. P0578 4341    M      ALF   20,CALL KRYTER 3-1270 ERRØR 1 $     $
      P0579 4C4C
     P057A 204B
     P057B 5259
     P057C 5445
     P057D 5220
     P057E 332D
     P057F 3132
     P0580 3730
     P0581 2045
     P0582 5252
     P0583 4F52
     P0584 2031
     P0585 2024
     P0586 2020
     P0587 2020
```

```
     P0588 2024
     P0589 2020
     P058A 2020
     P058B 2020
023. P058C 0000          NUM  0
024. P058D 5044    M2    ALF  14,PDP-10 A.Ø.K.
     P058E 502D
     P058F 3130
     P0590 2041
     P0591 2E4F
     P0592 2E4B
     P0593 2E20
     P0594 2020
     P0595 2020
     P0596 2020
     P0597 2020
     P0598 2020
     P0599 2020
     P059A 2020
025. P059B 0064          BZS  IBUF(INBUF)     ***HERE IS BUF.*****
 026. P05FF 7B05   XT1   ADC  TAB-MØM-1  BELØW IS DE-BUGG AID FØR
 027. P0600 0A31   DEBUG ENA  2*BUF+1  TYPING WHAT IS SENT TØ PDP10
 028. P0601 9800         SUB  PTR
     P0602 FB0A
029. P0603 680A          STA* MØM+5
030. P0604 C800          LDA  PTR
     P0605 FB07
031. P0606 88F8          ADD* XT1
032. P0607 6807          STA* MØM+6
033.               MØM   FWRITE $4,XMTR-MØM-1,,,A,5,5,,X
033. P0608 54F4
033. P0609 0D55
033. P060A 7B74
     P060B 0000
033. P060C 1004
033. P060D 0000
     P060E 0000
034. P060F 14EA          JMP- (D)
035. P0610 C800    DEBUG2 LDA  PTRP
     P0611 FC81
036. P0612 0F41          ARS  1
037. P0613 6806          STA* MD2+5
038.               MD2   FWRITE  $4,GG2-MD2-1,IBUF-MD2-1,,A,5,5,,X
 038. P0614 54F4
038. P0615 0D55
038. P0616 7C2A
     P0617 0000
038. P0618 1004
038. P0619 0000
     P061A 7F85
039. P061B 14EA          JMP- (D)
040.                     END
```

```
I        00FF  SYSTAT   7F4E  CØRSTA   006E  D        00EA  SYSLIB   00EB
 INBUF    0064  BUF      0018  FRY      7F41  AHEX     0089  ARE      0088
 ØNB      0023  ZR       0033  TIM      7F05  INA      7F00  START    0000P
 DIAL     000DP ENTRYS   0026P ITS      0027P HI2      0045P AS       0046P
 YE       0047P DØNE     0049P NØP      004FP DEL      005AP T6       005CP
 VALT     005FP DC       0061P QSV      0062P ASV      0063P BAC      0064P
 DAØT     0065P DCTØN    006AP DCTØK    0074P DCER     0081P HI       0087P
 NØT      0088P NLAT     008EP DISC     008FP STCRK    0095P ERMY     009AP
 AR       00A1P LAT      00A2P MSGT     00ACP INREL    00AFP FI       00B1P
 STRIN    00B2P STR2     00B4P NERV1    00BCP NØTBS    00BDP NØRR1    00C9P
 ERV      00D5P ØRR      00DAP RD       00DFP LSB      00E7P DRM      00EAP
 DRMWR    00EBP EXP      00EEP TAL2     00F3P DØR      0104P STXR     010CP
 PTR      010DP CKSUM    010EP TAB      010FP ØUTST    0150P ENIR     0151P
 IIHR     015AP ENXMT    0160P IINXM    0167P ØUTP     016DP DAØTT    017AP
 XMTR     017EP STXX     0187P MUC      018FP RLCTR    0195P FETCH    0196P
 LARUI    01A9P S1NØT    01AFP NSSP1    01B4P RETRY    01BEP KEEP     01C3P
 THIRE    01C5P ISSP1    01C6P S1       01C9P GØSTN    01CCP CR2      01D0P
 ERT      01D8P MSØUT    01D9P ØFFG     01DBP CR3      01DDP NS       01E3P
 ØVS1     01E4P NSS      01F1P ØVR      01F4P UNG      01F8P GØØD     0200P
 NØT1S    0204P EVEN     020EP TES      0212P TRB      0214P SENDØ    021BP
 NUS      021DP ØULP     0222P BRT      0223P TRYØ     0226P ØDD      0228P
 NR3      022EP DIS      0232P TDC      0233P NØTN     0236P SP1      023AP
 SP2      023BP YETØ     023CP GG2      0240P NT4      0247P AMLØG    024AP
 NKSF     0250P JB       025AP NØKSF    025CP NØLØGN   0267P T4       0269P
 NLIN     0270P CKCLØ    0276P SETTY    027FP NØTCLS   0282P YET2     0287P
 CR       028AP CRC      028BP PTRP     0293P LASCH    0294P TMDØT    0295P
 STPC     0296P LINN     0297P NULTA    02A7P NUL2     02AAP BA2      02AEP
 Z        02AFP LØØ      02B4P LØGI     02B5P ERNS     02BBP ØNE      02BFP
 GØNP     02C9P NØTNP    02CAP JMEM     02D9P ITPAS    02DAP NP       02E0P
 N10      02E1P NØT12    02E4P NNLA3    02E8P ERRMØ    02ECP JPL10    02EFP
 LA4      02F0P LA3      02FAP DMSG     0302P LA10     030AP NB76     030EP
 B7       031FP EM       0321P MSGG     0324P THREE    0328P NL9      032EP
 NIAC     0333P TIZ3     033BP NØT3     0340P LAS13    0341P IACC     0343P
 CKRE     0345P NIST5    034CP ERML     0351P TØF2     0353P LAS      0355P
 T10      0356P IST5     035CP ERM      035EP MMØØ1    0361P SETST    0363P
 CLRS     036CP LØPC     0374P NREL     0375P RELL     038EP DIALAD   0391P
 TWØ      0392P N8       0399P ERJ      039EP NQ       03A2P NY7      03A6P
 IN8      03AAP ANT      03B2P TTJØ     03B7P Y7       03B8P FA4F     03C1P
 SCH      03C4P T9       03C7P LØP7     03CCP NUP      03CDP ERF      03D1P
 TRYØFF   03D4P LØGØF    03D6P ITSREL   03E1P MØRE     03E2P REZ2     03E5P
 ITS4     03E6P P10      03EBP BUFI     03F6P ERMG     0401P THNØ     0405P
 SECRØ    041BP CKTT     041EP FØFØ     0427P TØT3     0428P ERK      0429P
 NRKK     042FP INEK     0439P ERKK     043EP KJT      0442P ERB      044AP
 SLØPC    044DP ERD      044EP S        0450P ERXX     0459P TRYØ3    045BP
 RN8      045DP ISN9     0462P NØTM10   0465P ERQQ     046EP KSLF     0471P
 TBØJ     047EP ITS10    047FP ERQ      0485P IS9      0489P NNØ5     049AP
 GØBØJ    04A3P ERE      04A8P FIVE     04ABP SIX      04B5P SEVEN    04B6P
 ERIR     04BAP LL10     04BDP ITPE     04C2P ASCH     04C4P CTLP     04C5P
 MSGGI    04C6P MSGK     04D5P MADD     04DDP RE7      04E4P FØUR     04E5P
 ITBØJ    04EEP CKIF     04F7P ERRP     04FBP TØØF     04FDP MEL      04FFP
 THS      050DP FIVF     050EP MMØØ     050FP TMØØ     0510P TM02     0512P
 ZNZ      051FP ZNZ1     0520P ADD      0529P FIN      052DP LP7      0533P
 LPS      0541P ME1      054BP ME2      054EP ME3      0552P ME4      0555P
 ME7      0558P ME8      055EP ME9      0567P ME10     056EP ME11     056FP
 ME12     0570P ME13     0574P MEMA     0576P M        0578P M2       058DP
 IBUF     059BP XT1      05FFP DEBUG    0600P MØM      0608P DEBUG2   0610P
 MD2      0614P DIALUP   0391X
 J
```

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

5.2 Programmed Response to PDP-10 Messages

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

Last MSG Sent to PDP-10	Response Received	Programmed Response for DIAL
LOGIN(6415,26)	JOB ##, etc. (2 or 4 lines)	Clear bit 5, Set Bit 2, Send: RXXXXX (Password)
	Please KJOB (2 or 4 lines)	Clear bit 5, Set bit 7, Type Error Msg-B, and Send: ↑C↑C K/F
	PASSWORD: (1 line)	Set bit 2, Send: RXXXXX (Password)
	None of the Above (2 lines)	Clear bit 5, Set bit 1, Type Error Msg-D and Send: SYSTAT[6415,26] If 3rd time, type error MSG-X and go to TRYOFF* exit.
ATT##[6415,26]	? can't, etc. (2 lines)	Clear bit 10, Set bit 1, Type error MSG-I and send: SYSTAT[6415,26] If 3rd time, type error MSG-X and go to TRYOFF.*
	? TTY, etc. (2 lines)	Clear bit 10, Type error MSG-C, Clear $ 7F4F Reset Status word to $40, and release.
	? ILLEG, etc. (2 lines)	Clear bit 10, Set bit 1, and Send: SYSTAT [6415, 26] If 3rd time, type error msg-x and go to TRYOFF*
	. (1 line)	Type error msg-B 2rd log-off by: Set bit 7, clear STXR, send: K/F If not 1st time, type error msg-Z and release
↑C↑CK/F	•Tota, etc. (5 lines)	Null $7F4F, reset status to $40, hang-up phone and release.
	other than above (5 lines)	Type error msg-R, go to TRYOFF*
	?NOT, etc. (2 lines)	Null $7F4F, reset status to $40, hang-up phone and release.
	CONFIRM (1 or 2 lines)	Send: F

Last MSG Sent to PDP-10	Response Received	Programmed Response for DIAL
	Other than above (2 lines)	Type Error msg-Q and go to TRYOFF*
	Saved, etc. -or- •JOB, etc. (3 lines)	Null $7F4F, reset status to $40, hang-up phone and release
	• (1 line)	Set bit 7, clear STXR, send: K/F If not 1st time, send error msg-Z and release.
	telephone disconnect (0-7 lines)	If logged in, (Bit15=1), & 1st Chars. are •Total or .Job, then set bit 10, Clear bits 5 & 15, and re-dial 3-1881.
↑C	any msg (1-7 lines)	If signing-off, (Bit 7=1), then send: ↑C↑C K/F Otherwise send: R SETTTY
	• (1 line)	Set bit 2, clear bits 2, 6, and 10. Send: ↑C
	####, etc. (>10 chrs.)	Save #### as time of day, Set bit 15, clear bits 2 & 6, send: ↑ϕ
	OTHE, etc. (>10 chrs.)	Clear bit 2, Set bit 1, Pause 8 secs and send: SYSTAT [6415,26]
RXXXXX (Password)		If 3rd time, type error msg-X, go to TRYOFF.*
	None of the above (>10 chrs.)	Clear bit 2, type error msg-G, go to TRYOFF*
	(<10 chrs. in 20 sec.)	Type error msg F, go to TRYOFF*
	• (1 line)	(indicates bad xfr) If bit 7 not set, send: RU INP If 3rd time type error msg-G and Logoff.
DONE XX	? HALT, etc. (2 lines)	Same as above
	None of the above (2 lines)	Type error msg-Q and go to TRYOFF*

Last MSG Sent to PDP-10	Response Received	Programmed Response for DIAL
any MSG	none in 60 secs.	Pause 1 min., if 3rd time this session, type error msg-S, and go to TRYOFF.*
SYS[6415,26]	##, etc. (2 or 3 lines)	Save ## as job no., Clear bit 1, set bit 10, Send: ATT ## [6415,26]
	.KJOB (2 or 3 lines)	Clear Bit 1. If bit 7 is set (signing-off) then clear $7F4F, set status to $40, hang-up and release. Otherwise set bit 5 and send: LOGIN (6415,26)
	none of the above (2 or 3 lines)	Clear bit 1, type error Msg-E go to TRYOFF*
↑ϕ	Any MSG (1 to 7 lines)	If bit 7 is set, (signing-off), send: ↑C↑C K/F Otherwise set bits 13, clear bit 11, reset LSB to $4800 and send: RU INP
	None in 60 sec.	After 3rd time type error msg-S and go to TRYOFF
RU INP	$2Λ (Binary) (1 line)	If $2A is followed by $19, reset counters, null input buffer and go to STRIN for next block XFR. Otherwise re-send last block. If 50th re-send type error msg-T & logoff.
	? (1 line)	When followed by H; re-iniate entire data XFR by sending: Ru INP (except when bit 7 is set which sends: ↑C↑C K/F
NONE	IACC, etc. (3 lines)	Clear bit 3. If bit-10 = 1; Send: ATT##[6415,26] Otherwise, if bit-5 = 1; send LOGIN (6415, 26), otherwise if bit-1 = 1; send: SYS[6415,26], otherwise type error msg-L and go to TRYOFF.*
	None	Disconnect and re-dial 3-1881 every 2 mins. If 5 hrs has elapsed, type error msg-1 and release.

*TRYOFF is an exit routine that goes to LOGOF if bit 15 = 0. Otherwise Bits 1,3,5,6,7,10,15 are retained, the phone is disconnected and a release is scheduled.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

5.3 FRYLIN-DIALUP Error MSG Code

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

Msg. Code	Meaning and Response
1	No connection after five attempts every 10 minutes for 5 hrs 5 hrs (will try again after 24:00).
2	Status indicates logged-in without a Job No. (will try getting status and attaching and if bit 7 is not set will run as normal entry after typing msg. - N).
3	Input buffer overflow or line count exceeds 7. However sequential carriage-returns are counted as one and $7F is ignored along with line-feed. (Response is to attempt log-off if this is not the first overflow, otherwise a control-0 is sent if the last char. sent was not a control-0. If the last was control-0, assume the overflow is due to failure of PDP-10 to recognize the char. and the "message of the day" has filled the buffer.
4	CDC-1700 Data-Control-Terminal (DCT) or DAC has failed (will try again on next noise scan).
5	No dial tone. (Try again next noise scan.)
6	Ru INP Halting re-try 3 times.
7	Status indicates a Job No. assigned but not logged-in. (Responds same as in 2 above.)
8	Low core status ($6E) disagrees with common status ($7F4E). (Assume low core is correct and try again.)
9	Attempting to Run Dialup while already logged-in with a reasonable Job No. (Try again next noise scan.)
A	Job No. found in $7F4F is >60 or <0. (Responds as in 2 above.)
B	Logging or attaching while already on or bad attach request (Log-off).
C	TTY already attached (reset status to $40 hang-up).
D	Response to LOGIN msg was not "JOB" nor "? please KJOB" (response same as 2 above).
E	Response to SYSTAT request other than numerals or "KJOB" (If logged-in, attempt to Log-off one time otherwise hand up and call back next scan cycle).
F	Did not receive at least 10 chars. within 20 secs of sending password. (Response same as E above).

Msg. Code	Meaning and Response
G	First chars. received after sending password are other than numerals or "OT." (Same as E.)
H	One word reply other than "." or "PASSWORD:" (Same as E)
I	Cannot attach to old Job No. (Same as 2 above.)
J	Two line response beginning "?C" not logical after 3 msg. repeats. (Same as E.)
K	Two line response to ATTACH request other than "?C", "?TTY" or "?I." or ?N or ?TOO FW. (Same as E).
L	Connected with "IACC etc." but not status word bits for Login, Attach, or status request (Same as E).
M	Three line response other than "IACC etc." or "CPU etc." (Same as E)
N	Chamber D.C. voltage bad.
O	One line response not logical (Same as E).
P	Four line response not following other than "DONE" msg. (Same as E first try, second try - release).
Q	Two line response not logical (Same as E).
R	Five, Six, or Seven line reply not logical. (Same as E.)
S	PDP-10 did not respond to last msg. (hang-up until next scan cycle.
T	Check sum errors exceeded 50 (Same as E).
U	Drum address for PSD data not within proper bounds ($4800 < LSB < $6800), (Don't call PDP-10 and don't run noise scan; call back next DT).
V	Data - Msg. string busy flag been set too long. Looping in STRIN. (Same as E.)
W	Same as V except looping is MMOO.
X	Status attempts exceeded 3 trys. (Same as E.)
Y	Status attempts exceeded 3 trys. (Same as E)
Z	2d try to log-off (hang-up)

ORNL/TM-5476
Dist. Category
UC-79 and 79m

INTERNAL DISTRIBUTION

1. R. K. Adams
2. J. L. Anderson
3. R. S. Booth
4. C. J. Borkowski
5-15. J. B. Bullock
16. M. K. Churchich
17. J. A. Cox
18. D. N. Fry
19-20. R. C. Gonzalez
21. R. C. Kryter
22. L. C. Oakes
23. G. S. Sadowski
24-28. W. H. Sides, Jr.
29. J. A. Watts
30. K. W. West
31-32. Central Research Library
33. Document Reference Section
34-37. Laboratory Records Department
38. Laboratory Records, ORNL R.C.
39. ORNL Patent Office

EXTERNAL DISTRIBUTION

40. Research and Technical Support Division, ERDA-ORO
41-279. Given distribution as shown in TID-4500 under UC-79m Category