RP11E

DISK PACK FORMATTER

MD-11-DZRP2-B

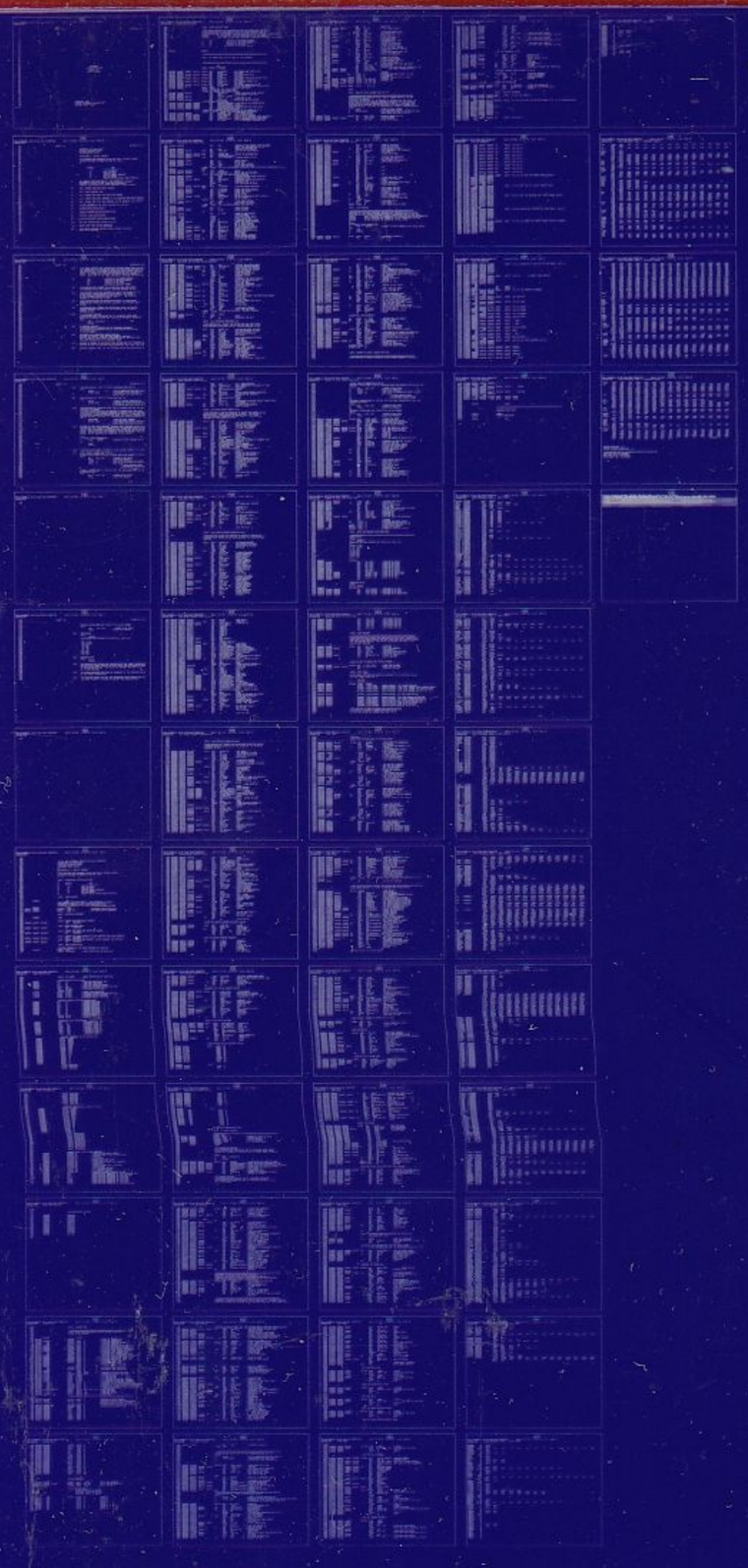
EP-DZRP2-B-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A



```
                        .REM    %
                        DZRP2   LST                                     02-APR-76 13:4
```

DOCUMENT
****
DZRP2 LST
****

PROGRAM BY C. HESS/F. ROEMER

THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.

15

SWITCH	USE
15	HALT ON ERROR
14	LOOP ON TEST
13	INHIBIT ERROR TYPEOUTS
12	BELL ON ERROR
7	INHIBIT DRIVE STATUS TABLE PRINTOUT

27 ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

38 200 = FORMAT DISK AND CHECK HEADERS

41 204 = CHECK HEADERS ONLY

44 210 = FORMAT AND CHECK THE SPECIFIED HEADER

47 214 = FORMAT AND CHECK HEADERS IN THE OPERATOR SPECIFIED SEQUENC

49 220 = CHANGE THE RP11E BUS ADDRESS, VECTOR ADDRESS, AND PRIORITY

54 INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***

59 MISCELLANEOUS DEFINITIONS

71 GENERAL PURPOSE REGISTER DEFINITIONS

83 PRIORITY LEVEL DEFINITIONS

93 "SWITCH REGISTER" SWITCH DEFINITIONS

121 DATA BIT DEFINITIONS (BIT00 TO BIT15)

149 BASIC "CPU" TRAP VECTOR ADDRESSES

190 THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
USED IN THE PROGRAM.

```
                                DZRP2  LST                                  02-APR-76 13:4


                        285     THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCU
                                THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE I
                                NOTE1:  IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                NOTE2:  EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS

                        291             EM              ;;POINTS TO THE ERROR MESSAGE
                                        DH              ;;POINTS TO THE DATA HEADER
                                        DT              ;;POINTS TO THE DATA
                                        DF              ;;POINTS TO THE DATA FORMAT

                        430     THIS ROUTINE WILL FORMAT THE ENTIRE PACK AND THEN VERIFY
                                THE HEADERS WITHIN EACH SECTOR. STARTING ADDRESS IS 200.

                        468     THIS ROUTINE ALLOWS THE OPERATOR TO VERIFY THE FORMAT OF THE
                                ENTIRE PACK.  THIS ROUTINE IS AUTOMATICALLY INITIATED
                                BY THE STANDARD FORMATTER ROUTINE (TEST1).  THE STARTING
                                ADDRESS OF THIS ROUTINE IS 204.

                        527     THIS ROUTINE ALLOWS THE OPERATOR TO SPECIFY A PARTICULAR
                                HEADER TO BE FORMATTED AND THEN VERIFIED. THE STARTING ADDR
                                IS 210.

                        623     THIS ROUTINE WILL FORMAT THE ENTIRE PACK USING THE SECTOR
                                ADDRESS SEQUENCE SPECIFIED BY THE OPERATOR. THE STARTING
                                ADDRESS IS 214.

                        821     TK INITIALIZE ROUTINE
                                THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
                                SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
                                CALL:
                                        JSR     PC,$TKINT
                                        RETURN

                        838     TK SERVICE ROUTINE
                                THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
                                BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
                                IT IN THE QUEUE.

                        882     SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
                                SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
                                CALL WHEN OPERATING IN TTY INTERRUPT MODE.

                        897     CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SE
                                ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RES

                        899     CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SE
```

DZRP2 LST 02-APR-76 13:4

```
                        960     THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                CALL:
                                        RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
                                        RETURN HERE             ;;CHARACTER IS ON THE STACK
                                                                ;;WITH PARITY BIT STRIPPED OFF

                        984     THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                CALL:
                                        RDLIN                   ;;INPUT A STRING FROM THE TTY
                                        RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WIL
                                                                ;;TERMINATOR WILL BE A BYTE OF A

                        1053    THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
                                CHANGE IT TO BINARY.
                                THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
                                OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPE
                                FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUS
                                THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RE
                                CALL:
                                        RDOCT                   ;;READ AN OCTAL NUMBER
                                        RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF T
                                                                ;;HIGH ORDER BITS ARE IN $HIOCT

                        1107    ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 B
                                THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE
                                NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CH
                                NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED
                                NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.

                                CALL:
                                1) USING A TRAP INSTRUCTION
                                        TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN
                                OR
                                        TYPE
                                        MESADR


                        1178    THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIG

                        1179    OCTAL (ASCII) NUMBER AND TYPE IT.
                                $TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS
                                CALL:
                                        MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        TYPOS                   ;;CALL FOR TYPEOUT
                                        .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS
                                        .BYTE   M               ;;M=1 OR 0
                                                                        ;;1=TYPE LEADING ZEROS
                                                                        ;;0=SUPPRESS LEADING ZER

                                $TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE
                                $TYPOS OR $TYPOC
                                CALL:
                                        MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        TYPON                   ;;CALL FOR TYPEOUT
```

```
208
```

```
                        ;TYPOC---ENTER HERE FOR TYPEOU  OF A 16 BIT NUMBER
                        ;CALL:
                        ;       MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                        ;       TYPOC                   ;;CALL FOR TYPEOUT

1256                    ;SAVE R0-R5
                        ;CALL:
                        ;       SAVREG
                        ;UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                        ;
                        ;TOP---(+16)
                        ; +2---(+18)
                        ; +4---R5
                        ; +6---R4
                        ; +8---R3
                        ;+10---R2
                        ;+12---R1
                        ;+14---R0

1283                    ;RESTORE R0-R5
                        ;CALL:
                        ;       RESREG

1302                    ;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTIO
                        ;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDR
                        ;OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                        ;GO TO THAT ROUTINE.

1324                    ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLE
                        ;BY THE "TRAP" INSTRUCTION.

1420                    ;THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVA
                        ;THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.
```

```
   245                                  %
```

```
                                .TITLE  MD-11-DZRP2-B, RP11E DISK PACK FORMATTER
                                ;*COPYRIGHT (C) 1975,1976
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY C. HESS/F. ROEMER
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C1),MAR 24, 1976.
                                ;*

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          ------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        10            BELL ON ERROR
                                ;*         7            INHIBIT DRIVE STATUS TABLE PRINTOUT

                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                                  .=174
000174  000000                DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

                                .SBTTL  STARTING ADDRESSES

000200                                  .=200

                                ;*200 = FORMAT DISK AND CHECK HEADERS
000200  000137  001304                  JMP     @#START1

                                ;*204 = CHECK HEADERS ONLY
000204  000137  001314                  JMP     @#START2

                                ;*210 = FORMAT AND CHECK THE SPECIFIED HEADER
000210  000137  001324                  JMP     @#START3

                                ;*214 = FORMAT AND CHECK HEADERS IN THE OPERATOR SPECIFIED SEQUENCE
000214  000137  001334                  JMP     @#START4
                                ;*220 = CHANGE THE RP11E BUS ADDRESS, VECTOR ADDRESS, AND PRIORITY
000220  000137  001344                  JMP     @#START5

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
```

```
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
```

```
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
            100000              BIT15=  100000
            040000              BIT14=  40000
            020000              BIT13=  20000
            010000              BIT12=  10000
            004000              BIT11=  4000
            002000              BIT10=  2000
            001000              BIT09=  1000
            000400              BIT08=  400
            000200              BIT07=  200
            000100              BIT06=  100
            000040              BIT05=  40
            000020              BIT04=  20
            000010              BIT03=  10
            000004              BIT02=  4
            000002              BIT01=  2
            000001              BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
                                .EQUIV  BIT07,BIT7
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
            000004              ERRVEC= 4               ;TIME OUT AND OTHER ERRORS
            000010              RESVEC= 10              ;RESERVED AND ILLEGAL INSTRUCTIONS
            000014              TPTVEC=14               ;"T" BIT
            000014              TRTVEC= 14              ;TRACE TRAP
            000014              BPTVEC= 14              ;BREAKPOINT TRAP (BPT)
            000020              IOTVEC= 20              ;INPUT/OUTPUT TRAP (IOT) **SCOPE**
            000024              PWRVEC= 24              ;POWER FAIL
            000030              EMTVEC= 30              ;EMULATOR TRAP (EMT) **ERROR**
            000034              TRAPVEC=34              ;"TRAP" TRAP
            000060              TKVEC=  60              ;TTY KEYBOARD VECTOR
            000064              TPVEC=  64              ;TTY PRINTER VECTOR
            000240              PIRQVEC=240             ;PROGRAM INTERRUPT REQUEST VECTOR
            000001              GB=1                    ;ERROR ROUTINE PARAMETER TO TYPE GOOD/BAD DATA
            000002              DD=2                    ;ERROR ROUTINE PARAMETER TO TYPE THE DATA
            000004              DS=4                    ;ERROR ROUTINE PARAMETER TO TYPE RPDS
            000010              ER=10                   ;ERROR ROUTINE PARAMETER TO TYPE RPER
            000020              CS=20                   ;ERROR ROUTINE PARAMETER TO TYPE RPCS
            000040              DA=40                   ;ERROR ROUTINE PARAMETER TO TYPE RPDA
            000100              HD=100                  ;ERROR ROUTINE PARAMETER TO TYPE THE HEADER
```

```
                                ;RP11E REGISTER INDEXES

000000                          RPDS=00
000002                          RPER=02
000004                          RPCS=04
000006                          RPWC=06
000010                          RPBA=10
000012                          RPCA=12
000014                          RPDA=14
000016                          RPM1=16
000020                          RPM2=20
000022                          RPM3=22
000024                          SUCA=24
000026                          SILO=26
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                    001100              .=1100
001100                          $CMTAG:                 ;;START OF COMMON TAGS
001100  000000      $PASS:  .WORD   0                   ;;CONTAINS PASS COUNT
001102  000         $TSTNM: .BYTE   0                   ;;CONTAINS THE TEST NUMBER
001103  000         $ERFLG: .BYTE   0                   ;;CONTAINS ERROR FLAG
001104  000000      $ICNT:  .WORD   0                   ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000      $LPADR: .WORD   0                   ;;CONTAINS SCOPE LOOP ADDRESS
001110  00000C      $LPERR: .WORD   0                   ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000      $ERTTL: .WORD   0                   ;;CONTAINS TOTAL ERRORS DETECTED
001114  000         $ITEMB: .BYTE   0                   ;;CONTAINS ITEM CONTROL BYTE
001115  001         $ERMAX: .BYTE   1                   ;;CONTAINS MAX. ERRORS PER TEST
001116  000000      $ERRPC: .WORD   0                   ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000      $GDADR: .WORD   0                   ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000      $BDADR: .WORD   0                   ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000      $GDDAT: .WORD   0                   ;;CONTAINS 'GOOD' DATA
001126  000000      $BDDAT: .WORD   0                   ;;CONTAINS 'BAD' DATA
001130  000000              .WORD   0                   ;;RESERVED--NOT TO BE USED
001132  000000              .WORD   0
001134  000         $AUTOB: .BYTE   0                   ;;AUTOMATIC MODE INDICATOR
001135  000         $INTAG: .BYTE   0                   ;;INTERRUPT MODE INDICATOR
001136  000000              .WORD   0
001140  177570      $SWR:   .WORD   DSWR                ;;ADDRESS OF SWITCH REGISTER
001142  177570      DISPLAY: .WORD  DDISP               ;;ADDRESS OF DISPLAY REGISTER
001144  177560      $TKS:   177560                      ;;TTY KBD STATUS
001146  177562      $TKB:   177562                      ;;TTY KBD BUFFER
001150  177564      $TPS:   177564                      ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566      $TPB:   177566                      ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000         $NULL:  .BYTE   0                   ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002         $FILLS: .BYTE   2                   ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012         $FILLC: .BYTE   12                  ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000         $TFFLG: .BYTE   0                   ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  177607  000377 $BELL:  .ASCIZ  <207><377><377>  ;;CODE FOR BELL
001164  077       $QUES:  .ASCII  /?/                  ;;QUESTION MARK
001165  015       $CRLF:  .ASCII  <15>                 ;;CARRIAGE RETURN
001166  000012    $LF:    .ASCIZ  <12>                 ;;LINE FEED
                                ;;*****************************************************************

001170  000000      MAXCYL: .WORD   0                   ;MAXIMUM CYLINDER ADDRESS
001172  000000      DRVTYP: .WORD   0                   ;CONTAINS A BIT FOR EACH RP03 DRIVE
                                                        ;BIT00 IS DRIVE 0, BIT01 IS DRIVE 1, ETC.
001174  000000      OPTION: .WORD   0                   ;CONTAINS A BIT FOR THE TEST SELECTED
001176  000000      BUSADR: .WORD   0                   ;CHANGE RP11E ADDRESS FLAG
001200  000000      GOOD:   .WORD   0
001202  000000      BAD:    .WORD   0
001204  000000      DATA:   .WORD   0
001206  000000      TOG1:   .WORD   0
001210  000000      TOG2:   .WORD   0
001212  000000      WORK:   .WORD   0
001214  000000      WORK1:  .WORD   0
001216  000000      WORK2:  .WORD   0
```

```
                                WORK3:  .WORD   0
                                UNIT:   .WORD   0
                                ATNB:   .WORD   0
                                CYL:    .WORD   0
                                HED:    .WORD   0
                                SEC:    .WORD   0
                                CYLR:   .WORD   0
                                HEDR:   .WORD   0
                                SECR:   .WORD   0
                                CYLA:   .WORD   0
                                HEDA:   .WORD   0
                                SECA:   .WORD   0
                                TTG:    .WORD   0
                                TEMP1:  .WORD   0
                                MASK:   .WORD   0
                                ROTOG:  .WORD   0
                                LERR:   .WORD   0
                                DEBUF:  .WORD   0


                                ;RP11E ADDRESSES

001264  176710                  RPADR:  .WORD   176710          ;RP11E BUS ADDRESS
001266  000254  000256          RPVEC:  .WORD   254,256         ;RP11E VECTOR ADDRESSES
001272  000240                  RPPRIO: .WORD   (5*32.)          ;RP11E PRIORITY

                                ;DRIVE STATUS TABLE
                                ;       DRVSTA>0, DRIVE IS ONLINE
                                ;       DRVSTA=0, DRIVE IS OFFLINE
                                ;       DRVSTA<0, DRIVE IS UNSAFE

001274  000                     DRVSTA: .BYTE   0               ;DRIVE 0
001275  000                             .BYTE   0               ;DRIVE 1
001276  000                             .BYTE   0               ;DRIVE 2
001277  000                             .BYTE   0               ;DRIVE 3
001300  000                             .BYTE   0               ;DRIVE 4
001301  000                             .BYTE   0               ;DRIVE 5
001302  000                             .BYTE   0               ;DRIVE 6
001303  000                             .BYTE   0               ;DRIVE 7
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF SITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT

001304                          $ERRTB:

                                ;;*****************************************************************

                                ;NOTE: THE ERROR TABLE IS NOT USED IN THIS PROGRAM.

                                ;;*****************************************************************

                                .SBTTL  STAR7 OF PROGRAM

001304  012737  000001  001174  START1: MOV     #1,OPTION       ;SET FORMAT AND CHECK FLAG
001312  000421                  BR      START           ;CONTINUE
001314  012737  000002  001174  START2: MOV     #2,OPTION       ;SET CHECK ONLY FLAG
001322  000415                  BR      START           ;CONTINUE
001324  012737  000004  001174  START3: MOV     #4,OPTION       ;SET FORMAT SPECIFIED SECTOR FLAG
001332  000411                  BR      START           ;CONTINUE
001334  012737  000010  001174  START4: MOV     #10,OPTION      ;SET SPECIAL FORMAT FLAT
001342  000405                  BR      START           ;CONTINUE
001344  012737  177777  001176  START5: MOV     #-1,BUSADR      ;SET CHANGE ADDRESS FLAG
001352  005037  001174          CLR     OPTION          ;CLEAR THE OTHER START OPTION FLAGS
001356  000005                  START:  RESET           ;CLEAR EVERYTHING
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001360  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001364  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001366  022706  001140                  CMP     #SWR,R6 ;;DONE?
001372  001374                  BNE     .-6             ;;LOOP BACK IF NO
001374  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001400  012737  010276  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001406  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001414  012737  006634  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001422  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001430  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001434  012737  001470  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001442  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001450  012737  177570  001142          MOV     #DOISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001456  022777  177777  177454          CMP     #-1,@SWR        ;  RY TO REFERENCE HARDWARE SWR
```

```
001464  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
001466  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
001470  012716  001476          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001474  000002                          RTI
001476  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001504  012737  000174  001142          MOV     #DISPREG,DISPLAY
001512  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001516  005227  177777                  INC     #-1             ;FIRST START ?
001522  001035                          BNE     1$              ;BR IF NOT
001524  104401  001532                  TYPE    ,68$            ;;TYPE ASCIZ STRING
001530  000432                          BR      67$             ;;GET OVER THE ASCIZ
                                ;;68$:  .ASCIZ  <15><12><12>/MAINDEC-11-DZRP2-B, RP11E DISK PACK FORMATTER/<15><12>
001616                          67$:
001616  104401  001165          1$:     TYPE    ,$CRLF          ;CR-LF
001622  004737  004576                  JSR     PC,$TKINT       ;SETUP THE TTY KEYBOARD
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
001626  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
001632  001006                          BNE     69$             ;;BRANCH IF YES
001634  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
001642  001005                          BNE     70$             ;;BRANCH IF NO
001644  104405                          GTSWR                   ;;GET SOFT-SWR SETTINGS
001646  000403                          BR      70$
001650  112737  000001  001134  69$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
001656                          70$:
001656  004737  006720                  JSR     PC,GETADR       ;CHECK THE RP11E ADDRESS
001662  004737  007230                  JSR     PC,RPINIT       ;CHECK THE SYSTEM
001666  005227  177777                  INC     #-1             ;FIRST START ?
001672  001404                          BEQ     SYSTAT          ;BR IF IT IS
001674  032777  000200  177236          BIT     #SW07,@SWR      ;INHIBIT SUBSYSTEM STATUS TYPEOUT ?
001702  001070                          BNE     SELTST          ;BR IF INHIBIT
001704                          SYSTAT:
001704  104401  001712                  TYPE    ,65$            ;;TYPE ASCIZ STRING
001710  000412                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <15><12>/DRIVE STATUS:/<15><12><12>
001736                          64$:
001736  005001                          CLR     R1              ;TABLE POINTER
001740  012702  000001                  MOV     #1,R2           ;DRIVE COUNTER
001744                          1$:
001744  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
                                                                ;;DRIVE NUMBER
001746  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
001750     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
001751     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
001752  105761  001274                  TSTB    DRVSTA(R1)      ;CHECK THE DRIVE'S STATUS
001756  100405                          BMI     2$              ;BR IF DRIVE UNSAFE
001760  001422                          BEQ     5$              ;BR IF DRIVE OFFLINE
001762  012737  012322  002044          MOV     #ONLINE,7$      ;ONLINE MESSAGE ADDRESS
001770  000403                          BR      3$              ;CHECK DRIVE TYPE
001772  012737  012341  002044  2$:     MOV     #UNSAFE,7$      ;UNSAFE MESSAGE ADDRESS
[illegible]  030237  001172          3$:     BIT     R2,DRVTYP       ;RP03 DRIVE ?
[illegible]  001004                          BNE     4$              ;BR IF RP03
[illegible]  012737  012377  002050          MOV     #RP02,8$        ;RP02 MESSAGE ADDRESS
002014  000412                          BR      6$              ;TYPE THE STATUS TABLE
002016  012737  012404  002050  4$:     MOV     #RP03,8$        ;RP03 MESSAGE ADDRESS
```

```
                                        BR      6$                      ;TYPEOUT THE STATUS TABLE
                                5$:     MOV     #OFFLIN,7$              ;OFFLINE MESSAGE ADDRESS
                                        MOV     #S1,8$                  ;DUMMY DRIVE TYPE MESSAGE
                                6$:     TYPE                            ;TYPE THE DRIVE'S STATUS
                                7$:     .WORD   0                       ;MESSAGE ADDRESS
                                        TYPE                            ;TYPE THE DRIVE TYPE
                                8$:     .WORD   0                       ;MESSAGE ADDRESS
                                        TYPE    $CRLF                   ;CR-LF
                                        INC     R1                      ;INCREMENT THE DRIVE POINTER
                                        ASLB    R2                      ;SHIFT THE DRIVE COUNTER
                                        BCC     1$                      ;BR IF NOT END
                                SELTST: MOV     RPADR,R4                ;LOAD RP11 ADDRESS
                                        BIT     #1,OPTION               ;STANDARD FORMAT AND VERIFY ?
                                        BEQ     1$                      ;BR IF NOT
                                        JMP     TEST1                   ;DO THE TEST
                                1$:     BIT     #2,OPTION               ;VERIFY THE PACK ?
                                        BEQ     2$                      ;BR IF NOT
                                        JMP     TEST2                   ;DO THE TEST
                                2$:     BIT     #4,OPTION               ;FORMAT AND VERIFY THE SPECIFIED HEADER
                                        BEQ     3$                      ;BR IF NOT
                                        JMP     TEST3                   ;DO THE TEST
                                3$:     BIT     #10,OPTION              ;SPECIAL FORMAT AND VERIFY
                                        BEQ     4$                      ;BR IF NOT
                                        JMP     TEST4                   ;DO THE TEST
                                4$:     TYPE    ,$CRLF
                                        TYPE    ,65$                    ;;TYPE ASCIZ STRING
                                        BR      64$                     ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <12>/NO TEST SELECTED - RESTART PROGRAM/
                                64$:
                                        TYPE    ,$CRLF
                                5$:     HALT                            ;HALT
                                        BR      5$                      ;INTERLOCK THE HALT


                                .SBTTL  **** STANDARD FORMATTER ****


                                ;*THIS ROUTINE WILL FORMAT THE ENTIRE PACK AND THEN VERIFY
                                ;*THE HEADERS WITHIN EACH SECTOR. STARTING ADDRESS IS 200.


                                TEST1:  MOV     #STACK,SP               ;INITIALIZE STACK POINTER
                                        CLR     PS                      ;SET THE PRIORITY TO ZERO
                                        JSR     PC,CLRP                 ;CLEAR THE CONTROLLER
                                IL1:    JSR     PC,UNIQ                 ;GET UNIT#
                                URDY1:  TYPE    .SAMES
                                        JSR     PC,RIN                  ;WAIT FOR KEYBOARD RESPONSE
                                MS2:    JSR     PC,HOME                 ;HOME SEEK
                                        MOV     #RPI,@RPVEC             ;RP11 VECTOR
                                        MOV     #1,RPWC(R4)             ;ALLOW DSH BREAKS
                                        MOVB    UNIT,RPCS+1(R4)         ;DRIVE ADDRESS
                                        BIS     #14000,RPCS(R4)         ;SET 'MODE' & 'HDR'
                                        MOVB    #103,RPCS(R4)           ;WRITE WITH INTERRUPT ENABLE
                                        CLR     PSW                     ;LOWER PRIORITY
                                IL2:    MOV     RPDA(R4),R3             ;SAVE ADDRESS
                                        MOV     #177770,TOG1            ;SETUP TIMEOUT
                                IL3:    CMP     R3,RPDA(R4)             ;HAS ADDRESS CHANGED?
                                        BNE     IL2                     ;YES - RECYCLE
                                        DEC     TOG1                    ;NO - HAS 100 MSEC. TIMED OUT?
```

```
696  002360  001372                              BNE     IL3                ;NO - KEEP CHECKING
697  002362  104074                              ERROR   DS!CS!ER!DA        ;DISK ADDRESS NOT CHANGING IN TIME
698  002364  000740                              BR      NS2                ;TRY AGAIN
699  002366  032764  100000  000004  RPI:       BIT     #BIT15,RPCS(R4)    ;ERROR?
700  002374  001404                              BEQ     2$                 ;NO
701  002376  104034                              ERROR   ER!CS!DS           ;ERROR DURING FORMAT GENERATION
702  002400  012716  002266          1$:         MOV     #NS2,(SP)          ;CHANGE RETURN ADDRESS
703  002404  000410                              BR      4$                 ;EXIT
704  002406  032764  000200  000004  2$:         BIT     #BIT07,RPCS(R4)    ;DONE?
705  002414  001002                              BNE     3$                 ;YES
706  002416  104034                              ERROR   DS!ER!CS           ;EXTRANEOUS INTERRUPT DURING FORMAT PASS
707  002420  000767                              BR      1$                 ;TRY AGAIN
708  002422  012716  002450          3$:         MOV     #STR4,(SP)         ;CHANGE RETURN ADDRESS
709  002426  000002                  4$:         RTI                        ;EXIT TO PASS 2
710
711                                  .SBTTL  **** PACK FORMAT VERIFY ****
712
713                                  ;*THIS ROUTINE ALLOWS THE OPERATOR TO VERIFY THE FORMAT OF THE
714                                  ;*ENTIRE PACK.  THIS ROUTINE IS AUTOMATICALLY INITIATED
715                                  ;*BY THE STANDARD FORMATTER ROUTINE (TEST1).  THE STARTING
716                                  ;*ADDRESS OF THIS ROUTINE IS 204.
717
718  002430  012706  001100          TEST2:  MOV     #STACK,SP
719  002434  005037  177776                  CLR     PS                 ;SET THE PRIORITY TO ZERO
720  002440  004737  011216                  JSR     PC,CLRP            ;CLEAR THE CONTROLLER
721  002444  004737  007426                  JSR     PC,UNIQ            ;GET THE UNIT NUMBER
722  002450  012737  000340  177776  STR4:   MOV     #PR7,PSW           ;LOCKOUT INTERRUPTS
723  002456  104401  011664                  TYPE    ,NSWMES
724  002462  004737  010242                  JSR     PC,RIN             ;WAIT FOR OPERATOR
725  002466  004737  007642          P2L1:   JSR     PC,HOME            ;HOME SEEK
726  002472  005037  001226                  CLR     CYL                ;CYL=0
727  002476  013764  001226  000012  P2L2:   MOV     CYL,RPCA(R4)       ;START AT
728  002504  005064  000014                  CLR     RPDA(R4)           ;SAK & TAR = 0
729  002510  012764  176650  000006          MOV     #-600.,RPWC(R4)    ;READ ONE CYLINDER OF HEADERS INTO
730  002516  012764  012412  000010          MOV     #INBUF,RPBA(R4)    ;BUFFER
731  002524  113764  001222  000005          MOVB    UNIT,RPCS+1(R4)    ;DRIVE ADDRESS
732  002532  052764  014000  000004          BIS     #14000,RPCS(R4)    ;10/15 HEADER MODE
733  002540  112764  000005  000004          MOVB    #5,RPCS(R4)        ;READ HEADERS
734  002546  032764  100200  000004  1$:     BIT     #BIT15!BIT07,RPCS(R4)   ;ERROR OR DONE ?
735  002554  001774                          BEQ     1$                 ;NOT YET
736  002556  100004                          BPL     P2N1               ;DONE
737  002560  104074                          ERROR   ER!DS!CS!DA        ;ERROR ON READ HEADER OP.
738  002562  004737  011216                  JSR     PC,CLRP            ;CLEAR THE CONTROLLER
739  002566  000743                          BR      P2L2               ;RECYCLE
740  002570  005037  001230          P2N1:   CLR     HED                ;START ADDRESS CHECKING
741  002574  005037  001232                  CLR     SEC                ;INITIALIZE GOOD DATA
742  002600  012703  012412                  MOV     #INBUF,R3          ;POINTER TO DATA READ
743  002604  004737  007760          P2L3:   JSR     PC,CHKAD           ;CHECK DATA
744  002610  000240                          NOP                        ;NO LOOPING
745  002612  005237  001232                  INC     SEC                ;ADVANCE SEC#
746  002616  023727  001232  000011          CMP     SEC,#9.            ;OVFLO?
747  002624  101002                          BHI     1$                 ;YES
748  002626  000137  002604                  JMP     P2L3               ;NO- CHECK NEXT ADDRESS
749  002632  005037  001232          1$:     CLR     SEC
750  002636  005237  001230                  INC     HED                ;ADVANCE HEAD#
751  002642  023727  001230  000023          CMP     HED,#19.           ;OVFLO?
```

```
   752  002650  101002                          BHI     2$              ;YES
   753  002652  000137  002604                  JMP     P2L3            ;NO- CHECK NEXT ADDRESS
   754  002656  005037  001230          2$:     CLR     HED
   755  002662  005237  001226                  INC     CYL             ;ADVANCE CYL#
   756  002666  023737  001226  001170          CMP     CYL,MAXCYL      ;DONE ALL?
   757  002674  101002                          BHI     3$              ;YES.
   758  002676  000137  002476                  JMP     P2L2            ;NO- READ ANOTHER BLOCK
   759  002702  104401  012251          3$:     TYPE    ,ENDM1
   760  002706  032777  040000  176224          BIT     #SW14,@SWR      ;DO THE TEST AGAIN
   761  002714  001407                          BEQ     5$              ;BR IF NOT
   762  002716  032737  000001  001174          BIT     #1,OPTION       ;IN HERE FROM TEST 1 ?
   763  002724  001402                          BEQ     4$              ;BR IF NOT
   764  002726  000137  002256                  JMP     URDY1           ;DO TEST 1 AGAIN
   765  002732  000755                  4$:     BR      P2L1            ;DO TEST 2 AGAIN
   766  002734  000000                  5$:     HALT
   767  002736  000775                          BR      4$              ;LOCK THE HALT
   768
   769
   770                                  .SBTTL  **** SINGLE HEADER FORMATTER ****
   771
   772                                  ;*THIS ROUTINE ALLOWS THE OPERATOR TO SPECIFY A PARTICULAR
   773                                  ;*HEADER TO BE FORMATTED AND THEN VERIFIED. THE STARTING ADDR
   774                                  ;*IS 210.
   775
   776  002740  012706  001100          TEST3:  MOV     #STACK,SP       ;SET STACK POINTER
   777  002744  005037  177776                  CLR     PS              ;SET PRIORITY TO ZERO
   778  002750  004737  011216                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
   779  002754  004737  007426                  JSR     PC,UNIQ         ;GET UNIT
   780  002760  104401  012010                  TYPE    ,HDMS
   781  002764  104401  012077          HL1:    TYPE    ,OLDMS
   782  002770  013705  001170                  MOV     MAXCYL,R5       ;HIGHEST CYLINDER
   783  002774  004737  007570                  JSR     PC,PARIN        ;FETCH PARAMETER
   784  003000  000771                          BR      HL1             ;ERROR RECYCLE
   785  003002  013737  001214  001242          MOV     WORK1,CYLA      ;SAVE VALUE
   786  003010  012705  000023                  MOV     #19.,R5         ;HIGHEST TRACK
   787  003014  004737  007570                  JSR     PC,PARIN        ;FETCH PARAMETER
   788  003020  000761                          BR      HL1             ;ERROR RECYCLE
   789  003022  013737  001214  001244          MOV     WORK1,HEDA      ;SAVE VALUE
   790  003030  004737  010122                  JSR     PC,TTIO         ;FETCH PARAMETER
   791  003034  000753                          BR      HL1             ;ERROR RECYCLE
   792  003036  020027  000011                  CMP     R0,#9.          ;HIGHEST SECTOR
   793  003042  101350                          BHI     HL1             ;TOO BIG
   794  003044  010037  001246                  MOV     R0,SECA         ;SAVE VALUE
   795  003050  104401  012113          HL2:    TYPE    ,NEWMS
   796  003054  013705  001170                  MOV     MAXCYL,R5       ;HIGHEST CYLINDER
   797  003060  004737  007570                  JSR     PC,PARIN        ;FETCH PARAMETER
   798  003064  000771                          BR      HL2             ;ERROR RECYCLE
   799  003066  013737  001214  001226          MOV     WORK1,CYL       ;SAVE VALUE
   800  003074  012705  000023                  MOV     #19.,R5         ;HIGHEST TRACK
   801  003100  004737  007570                  JSR     PC,PARIN        ;FETCH PARAMETER
   802  003104  000761                          BR      HL2             ;ERROR RECYCLE
   803  003106  013737  001214  001230          MOV     WORK1,HED       ;SAVE VALUE
   804  003114  004737  010122                  JSR     PC,TTIO         ;FETCH PARAMETER
   805  003120  000753                          BR      HL2             ;ERROR RECYCLE
   806  003122  020027  000011                  CMP     R0,#9.          ;HIGHEST SECTOR
   807  003126  101350                          BHI     HL2             ;TOO BIG
```

```
808  003130  010037  001232                HGO:    MOV     R0,SEC          ;SAVE VALUE
809  003134  005037  012412                        CLR     INBUF           ;HWRD1=0
810  003140  013737  001232  012416                MOV     SEC,INBUF+4     ;HWRD3=SECTOR
811  003146  013737  001226  012414                MOV     CYL,INBUF+2
812  003154  006137  012414                       ROL     INBUF+2
813  003160  006137  012414                       ROL     INBUF+2
814  003164  006137  012414                       ROL     INBUF+2
815  003170  006137  012414                       ROL     INBUF+2
816  003174  006137  012414                       ROL     INBUF+2
817  003200  006137  012414                       ROL     INBUF+2
818  003204  006137  001230                       ROL     HED
819  003210  053737  001230  012414               BIS     HED,INBUF+2
820  003216  006037  001230                       ROR     HED
821  003222  042737  100001  012414               BIC     #100001,INBUF+2
822  003230  004737  007642                HGL1:   JSR     PC,HOME         ;HOME SEEK
823  003234  012764  177775  000006                MOV     #-3,RPWC(R4)    ;WORD COUNT
824  003242  012764  012412  000010                MOV     #INBUF,RPBA(R4) ;BUFFER ADDRESS
825  003250  113764  001222  000005                MOVB    UNIT,RPCS+1(R4) ;LOAD DRIVE NUMBER
826  003256  052764  014000  000004                BIS     #14000,RPCS(R4) ;10/15 MODE HEADER OP
827  003264  013764  001242  000012                MOV     CYLA,RPCA(R4)   ;LOAD CYLINDER
828  003272  113764  001244  000015                MOVB    HEDA,RPDA+1(R4) ;TRACK,
829  003300  113764  001246  000014                MOVB    SECA,RPDA(R4)   ;AND SECTOR TO BE CHANGED
830  003306  112764  000003  000004                MOVB    #3,RPCS(R4)     ;REWRITE IT
831  003314  032764  100200  000004  1$:           BIT     #BIT15!BIT07,RPCS(R4)   ;ERROR OR DONE?
832  003322  001774                                BEQ     1$              ;NOT YET
833  003324  100004                                BPL     HGO1            ;DONE
834  003326  104074                                ERROR   ER!DS!CS!DA     ;ERROR ON WRITE HEADER OP
835  003330  004737  011216                        JSR     PC,CLRP         ;CLEAR THE CONTROLLER
836  003334  000735                                BR      HGL1            ;TRY AGAIN
837  003336  005037  012412                HGO1:   CLR     INBUF           ;DESTROY
838  003342  005037  012414                        CLR     INBUF+2         ;OLD
839  003346  005037  012416                        CLR     INBUF+4         ;DATA
840  003352  004737  007642                HGL2:   JSR     PC,HOME         ;HOME SEEK
841  003356  012764  177775  000006                MOV     #-3,RPWC(R4)    ;WORD COUNT
842  003364  012764  012412  000010                MOV     #INBUF,RPBA(R4) ;BUFFER
843  003372  013764  001242  000012                MOV     CYLA,RPCA(R4)   ;DISK
844  003400  113764  001244  000015                MOVB    HEDA,RPDA+1(R4) ;ADDRESS
845  003406  113764  001246  000014                MOVB    SECA,RPDA(R4)   ;TO BE FOUND
846  003414  113764  001222  000005                MOVB    UNIT,RPCS+1(R4) ;LOAD DRIVE NUMBER
847  003422  052764  014000  000004                BIS     #14000,RPCS(R4) ;10/15 MODE HEADER OP
848  003430  112764  000005  000004                MOVB    #5,RPCS(R4)     ;READ
849  003436  032764  100200  000004  1$:           BIT     #BIT15!BIT07,RPCS(R4)   ;ERROR OR DONE?
850  003444  001774                                BEQ     1$              ;NOT YET
851  003446  100004                                BPL     HGO2            ;DONE
852  003450  104074                                ERROR   ER!DS!CS!DA     ;ERROR ON READING NEW HEADER
853  003452  004737  011216                        JSR     PC,CLRP         ;CLEAR THE CONTROLLER
854  003456  000735                                BR      HGL2            ;TRY AGAIN
855  003460  012703  012412                HGO2:   MOV     #INBUF,R3       ;R3=DATA POINTER
856  003464  004737  007760                        JSR     PC,CHKAD        ;CHECK REWRITTEN HEADER
857  003470  000730                                BR      HGL2            ;ERROR - TRY AGAIN
858  003472  104401  012251                        TYPE    ,ENDM1
859  003476  032777  040000  175434                BIT     #SW14,@SWR      ;DO THE TEST AGAIN ?
860  003504  001402                                BEQ     1$              ;BR IF NOT
861  003506  000137  003134                        JMP     HGO             ;TEST 3 AGAIN
862  003512  000000                        1$:     HALT
863  003514  000776                                BR      1$              ;LOCK THE HALT
```

```
                                        .SBTTL  **** SPECIAL FORMATTER ****

                                        ;*THIS ROUTINE WILL FORMAT THE ENTIRE PACK USING THE SECTOR
                                        ;*ADDRESS SEQUENCE SPECIFIED BY THE OPERATOR. THE STARTING
                                        ;*ADDRESS IS 214.

003516  012706  001100          TEST4:  MOV     #STACK,SP       ;SET STACK
003522  005037  177776                  CLR     PS              ;SET PRIORITY TO ZERO
003526  004737  011216                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
003532  004737  007426                  JSR     PC,UNIQ         ;GET UNIT#
003536  104401  011251          SFSL1:  TYPE    ,SECMES
003542  004737  004242                  JSR     PC,CLRS         ;CLEAR FLAG STRING
003546  012737  177766  001206          MOV     #-10.,TOG1      ;DO BELOW 10 TIMES
003554  012705  004504                  MOV     #MSTR,R5        ;POINTER TO MESSAGE POINTERS
003560  012703  004530                  MOV     #SCSLT,R3
003564  012500                  SFSL2:  MOV     (R5)+,R0        ;FETCH POINTER
003566  004737  011204                  JSR     PC,TYPOUT       ;FOR MESSAGE
003572  004737  004262                  JSR     PC,SIN          ;GET PARAMETER
003576  000757                          BR      SFSL1           ;ERROR
003600  005237  001206                  INC     TOG1            ;DONE?
003604  001367                          BNE     SFSL2           ;NOT YET
003606  004737  004332          SFSL3:  JSR     PC,MKBUF        ;GENERATE A TABLE OF ADDRESSES
003612  004737  007642                  JSR     PC,HOME         ;RESET SELECTED UNIT
003616  005037  001226                  CLR     CYL             ;START AT CYL=0
003622  013764  001226  000012  SFSL4:  MOV     CYL,RPCA(R4)    ;LOAD
003630  005064  000014                  CLR     RPDA(R4)                ;DISK ADDRESS
003634  004737  004430                  JSR     PC,SEKE         ;SEEK
003640  000762                          BR      SFSL3           ;ERROR RETURN
003642  012764  176650  000006          MOV     #-600.,RPWC(R4) ;GOOD RETURN
003650  012764  014712  000010          MOV     #OUTBUF,RPBA(R4)        ;WC&CA
003656  113764  001222  000005          MOVB    UNIT,RPCS+1(R4) ;LOAD DRIVE NUMBER
003664  052764  014000  000004          BIS     #14000,RPCS(R4) ;10/15 HEADER MODE
003672  112764  000003  000004          MOVB    #3,RPCS(R4)     ;WRITE
003700  032764  100200  000004  1$:     BIT     #BIT15!BIT07,RPCS(R4)   ;ERROR OR DONE
003706  001774                          BEQ     1$              ;WAIT
003710  100004                          BPL     SFSN1           ;DONE
003712  104074                          ERROR   ER!DS!CS!DA     ;ERROR ON WRITE HEADER OP
003714  004737  011216                  JSR     PC,CLRP         ;CLEAR THE CONTROLLER
003720  000740                          BR      SFSL4           ;REPEAT
003722  012737  177470  001206  SFSN1:  MOV     #-200.,TOG1
003730  012705  014714                  MOV     #OUTBUF+2,R5
003734  062715  000100          SFSL5:  ADD     #100,(R5)       ;ADVANCE CYLINDER ADDRESS
003740  062705  000006                  ADD     #6,R5           ;INDEX
003744  005237  001206                  INC     TOG1            ;DONE?
003750  001371                          BNE     SFSL5           ;NO
003752  005237  001226                  INC     CYL             ;ADVANCE CYLINDER
003756  023737  001226  001170          CMP     CYL,MAXCYL      ;DONE?
003764  101001                          BHI     SFCHK           ;YES
003766  000715                          BR      SFSL4           ;NO
003770  012737  000340  177776  SFCHK:  MOV     #PR7,PSW        ;LOCKOUT INTERRUPTS
003776  004737  007642                  JSR     PC,HOME         ;HOME SEEK
004002  005037  001226                  CLR     CYL             ;CYL=0
004006  013764  001226  000012  SFCL1:  MOV     CYL,RPCA(R4)    ;TAR=0
004014  005064  000014                  CLR     RPDA(R4)                ;SAR=0
```

```
                                        JSR     PC,SEKE         ;SEEK
                                        BR      SFCHK           ;ERROR RETURN
                                        MOV     #-600.,RPWC(R4) ;WC
                                        MOV     #INBUF,RPBA(R4) ;CA
                                        MOVB    UNIT,RPCS+1(R4) ;LOAD DRIVE NUMBER
                                        BIS     #14000,RPCS(R4) ;ID/IS HEADER OP
                                        MOVB    #5,RPCS(R4)     ;READ
                                1$:     BIT     #BIT15!BIT07,RPCS(R4)   ;ERROR OR DONE?
                                        BEQ     1$              ;NOT YET
                                        BPL     SFCN1           ;JUMP IF DONE
                                        ERROR   ER!DS!CS!DA     ;READ HEADER ERROR
                                        JSR     PC,CLRP         ;CLEAR THE CONTROLLER
                                        BR      SFCL1           ;LOOP
                                SFCN1:  MOV     #SCSLT,R5       ;SETUP SECTOR LIST AND
                                        MOV     #-10.,TOG1      ;COUNTER
                                        CLR     HED             ;SETUP GOOD DATA
                                        MOV     #INBUF,R3       ;SETUP BUFFER POINTER
                                SFCL2:  MOV     (R5)+,SEC       ;FETCH A SECTOR
                                        JSR     PC,CHKAD        ;CHECK ADDRESS
                                        NOP                     ;DON'T USE ERROR RETURN
                                        INC     TOG1            ;DONE ONE SURFACE?
                                        BNE     SFCL2           ;NO
                                        MOV     #SCSLT,R5       ;YES - RESET
                                        MOV     #-10.,TOG1      ;SECTOR PARAMETERS
                                        INC     HED             ;ADVANCE TRACK
                                        CMP     HED,#19.        ;DONE?
                                        BLOS    SFCL2           ;NO
                                        CLR     HED             ;YES - RESET HEAD
                                        INC     CYL             ;ADVANCE CYLINDER
                                        CMP     CYL,MAXCYL      ;DONE?
                                        BLOS    SFCL1           ;NOT YET
                                        TYPE    ,ENDM1
                                        BIT     #SW11,@SWR      ;DO THE TEST AGAIN ?
                                        BEQ     1$              ;BR IF NOT
                                        JMP     SFSL3           ;DO TEST 4 AGAIN
                                1$:     HALT
                                        BR      1$              ;LOCK THE HALT

                                ;CLEAR THE SECTOR ADDRESS SEQUENCE TABLES

                                CLRS:   MOV     #-30.,R5        ;CLEAR
                                        MOV     #SCSLT,R3       ;THE SECTOR
                                1$:     CLRB    (R3)+           ;SLOTS AND
                                        INC     R5              ;FLAG POINTERS
                                        BNE     1$              ;EXIT
                                        RTS     PC              ;WHEN DONE

                                ;GET THE SECTOR ADDRESS SEQUENCE

                                SIN:    JSR     PC,RIN          ;ASSEMBLE
                                        JSR     PC,TTO          ;ECHO
                                        SUB     #60,R0          ;A LIST OF
                                        CMP     R0,#9.          ;LOGICAL SECTOR
                                        BHI     1$              ;NUMBERS
                                        TST     R0              ;TOO SMALL?
                                        BMI     1$              ;0-9 ARE LEGAL
```

```
  976  004310  105760  004554                TSTB    SCFLG(R0)       ;HAS THIS NUMBER BEEN USED?
  977  004314  001005                        BNE     1$              ;ERROR - THIS NUMBER ALREADY USED
  978  004316  105160  004554                COMB    SCFLG(R0)       ;SET FLAG
  979  004322  010023                        MOV     R0,(R3)+        ;STORE SEC#
  980  004324  062716  000002                ADD     #2,(SP)         ;ADVANCE RETURN
  981  004330  000207                1$:     RTS     PC              ;AND EXIT
  982
  983                                ;GENERATE THE SECTOR ADDRESS BUFFER
  984
  985  004332  012737  177754  001206 MKBUF: MOV     #-20.,TOG1      ;TOG1=COUNTER
  986  004340  012705  014712                MOV     #OUTBUF,R5      ;R5=POINTER TO BUFFER
  987  004344  012703  004530                MOV     #SCSLT,R3       ;R3=POINTER TO SECTOR LIST
  988  004350  012737  177766  001210        MOV     #-10.,TOG2      ;TOG2=SECTOR COUNT
  989  004356  005037  001212                CLR     WORK
  990  004362  005025                1$:     CLR     (R5)+           ;WORD1=0
  991  004364  013725  001212                MOV     WORK,(R5)+      ;WORK2=CYL+HEAD
  992  004370  012325                        MOV     (R3)+,(R5)+     ;WORD3=SECTOR
  993  004372  005237  001210                INC     TOG2            ;DONE 10?
  994  004376  001371                        BNE     1$              ;NO
  995  004400  012703  004530                MOV     #SCSLT,R3       ;YES - UPDATE POINTER,
  996  004404  012737  177766  001210        MOV     #-10.,TOG2      ;COUNTER,
  997  004412  062737  000002  001212        ADD     #2,WORK         ;AND TRACK
  998  004420  005237  001206                INC     TOG1            ;FINISHED?
  999  004424  001356                        BNE     1$              ;NO
 1000  004426  000207                        RTS     PC              ;EXIT
 1001
 1002                                ;SEEK TO THE NEXT CYLINDER
 1003
 1004  004430  112764  000377  000000 SEKE:  MOVB    #377,RPDS(R4)   ;CLEAR ATTN
 1005  004436  112764  000011  000004        MOVB    #11,RPCS(R4)    ;SEEK
 1006  004444  033764  001224  000000 1$:    BIT     ATNB,RPDS(R4)   ;WAIT FOR
 1007  004452  001774                        BEQ     1$              ;ATTN
 1008  004454  112764  000377  000000        MOVB    #377,RPDS(R4)   ;CLEAR ATTN
 1009  004462  032764  004000  000000        BIT     #BIT11,RPDS(R4) ;DONE?
 1010  004470  001402                        BEQ     2$
 1011  004472  104074                        ERROR   ER!CS!DS!DA     ;SEEK INCOMPLETE
 1012  004474  000207                        RTS     PC              ;ERROR EXIT
 1013  004476  062716  000002        2$:     ADD     #2,(SP)         ;GOOD
 1014  004502  000207                        RTS     PC              ;EXIT
 1015
 1016                                ;USER DEFINED SECTOR SEQUENCE TABLES
 1017
 1018  004504  011341                NSTR:   .WORD   N0
 1019  004506  011347                        .WORD   N1
 1020  004510  011355                        .WORD   N2
 1021  004512  011363                        .WORD   N3
 1022  004514  011371                        .WORD   N4
 1023  004516  011377                        .WORD   N5
 1024  004520  011405                        .WORD   N6
 1025  004522  011413                        .WORD   N7
 1026  004524  011421                        .WORD   N8
 1027  004526  011427                        .WORD   N9
 1028
 1029  004530                        SCSLT:
 1030  004530  000000                        .WORD   0
 1031  004532  000000                        .WORD   0
```

```
1032  004534  000000                          .WORD   0
1033  004536  000000                          .WORD   0
1034  004540  000000                          .WORD   0
1035  004542  000000                          .WORD   0
1036  004544  000000                          .WORD   0
1037  004546  000000                          .WORD   0
1038  004550  000000                          .WORD   0
1039  004552  000000                          .WORD   0
1040
1041  004554                          SCFLG:
1042  004554      000                         .BYTE   0
1043  004555      000                         .BYTE   0
1044  004556      000                         .BYTE   0
1045  004557      000                         .BYTE   0
1046  004560      000                         .BYTE   0
1047  004561      000                         .BYTE   0
1048  004562      000                         .BYTE   0
1049  004563      000                         .BYTE   0
1050  004564      000                         .BYTE   0
1051  004565      000                         .BYTE   0
1052
1053                                  .SBTTL  #### SUBROUTINES ####
1054
1055                                  .SBTTL  TTY INPUT ROUTINE
1056
1057                                  ;;*******************************************************************
1058                                  .ENABL  LSB
1059  004566  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
1060  004570  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
1061  004572  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
1062  004574  000001                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
1063          004575                  $TKQEND=.
1064          004576                  .EVEN
1065
1066                                  ;*TK INITIALIZE ROUTINE
1067                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
1068                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
1069                                  ;
1070                                  ;*CALL:
1071                                  ;*      JSR     PC,$TKINT
1072                                  ;*      RETURN
1073                                  ;
1074  004576  005037  004566          $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
1075  004602  012737  004574  004570          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
1076  004610  013737  004570  004572          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
1077  004616  012737  004646  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
1078  004624  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
1079  004632  005777  174310                  TST     @$TKB           ;;CLEAR DONE FLAG
1080  004636  012777  000100  174300          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
1081  004644  000207                          RTS     PC              ;;RETURN TO CALLER
1082
1083                                  ;*TK SERVICE ROUTINE
1084                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
1085                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
1086                                  ;*IT IN THE QUEUE.
1087                                  ;
```

```
1088  004646  117746  174274          STKSRV: MOVB    @STKB,-(SP)     ;;PICKUP THE CHARACTER
1089  004652  042716  177600                  BIC     #^C177,(SP)     ;;STRIP THE JUNK
1090  004656  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
1091  004662  001004                          BNE     2$              ;;BRANCH IF NO
1092  004664  022737  000176  001140          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
1093  004672  001500                          BEQ     6$              ;;GO TO SWR CHANGE
1094
1095  004674                          2$:
1096  004674  022737  000001  004566          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
1097  004702  001004                          BNE     3$              ;;BRANCH IF NO
1098  004704  104401  001160                  TYPE    ,$BELL          ;;RING THE TTY BELL
1099  004710  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
1100  004712  000451                          BR      5$              ;;EXIT
1101  004714  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
1102  004720  001021                          BNE     32$             ;;BRANCH IF NO
1103  004722  005077  174216                  CLR     @STKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
1104  004726  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
1105  004730  105777  174210          31$:    TSTB    @STKS           ;;WAIT FOR A CHAR
1106  004734  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
1107  004736  117746  174204                  MOVB    @STKB,-(SP)     ;;GET THE CHARACTER
1108  004742  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
1109  004746  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
1110  004752  001366                          BNE     31$             ;;BRANCH IF NO
1111  004754  012777  000100  174162          MOV     #100,@STKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
1112  004762  000002                          RTI                     ;;RETURN
1113  004764  005237  004566          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
1114  004770  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
1115  004774  002405                          BLT     4$              ;;BRANCH IF YES
1116  004776  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
1117  005002  003002                          BGT     4$              ;;BRANCH IF YES
1118  005004  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
1119  005010  112677  177554          4$:     MOVB    (SP)+,@STKQIN   ;;AND PUT IT IN QUEUE
1120  005014  005237  004570                  INC     $TKQIN          ;;UPDATE THE POINTER
1121  005020  023727  004570  004575          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
1122  005026  001003                          BNE     5$              ;;BRANCH IF NO
1123  005030  012737  004574  004570          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
1124  005036  000002                  5$:     RTI                     ;;RETURN
1125
1126                                  ;;*****************************************************************
1127                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
1128                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
1129                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
1130                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
1131  005040  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED
1132  005046  001104                          BNE     15$             ;;EXIT IF NOT
1133  005050  105777  174070                  TSTB    @STKS           ;;IS A CHAR WAITING?
1134  005054  100101                          BPL     15$             ;;IF NOT, EXIT
1135  005056  117746  174064                  MOVB    @STKB,-(SP)     ;;YES
1136  005062  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT  7-BIT ASCII
1137  005066  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
1138  005072  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
1139                                                                  ;;AND EXIT
1140
1141                                  ;;*****************************************************************
1142                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
1143                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
```

```
1144                                                ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
1145  005074  123727  001134  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
1146  005102  001674                        BEQ     2$              ;;BRANCH IF YES
1147  005104  005726                        TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
1148  005106  004737  004576                JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
1149  005112  005077  174026                CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
1150  005116  112737  000001  001135        MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
1151
1152  005124  104401  005702                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (↑G)
1153  005130  104401  005707        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
1154  005134  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
1155  005140  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1156  005142  104401  005720                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
1157  005146  005046                19$:     CLR     -(SP)           ;;CLEAR COUNTER
1158  005150  005046                        CLR     -(SP)           ;;THE NEW SWR
1159  005152  105777  173766        7$:     TSTB    @$TKS           ;;CHAR THERE?
1160  005156  100375                        BPL     7$              ;;IF NOT TRY AGAIN
1161
1162  005160  117746  173762                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
1163  005164  042716  177600                BIC     #↑C177,(SP)     ;;MAKE IT 7-BIT ASCII
1164
1165
1166
1167  005170  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
1168  005174  001005                        BNE     10$             ;;BRANCH IF NOT
1169  005176  104401  005675                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (↑U)
1170  005202  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
1171  005206  000757                        BR      19$             ;;LET'S TRY IT AGAIN
1172
1173
1174  005210  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
1175  005214  001022                        BNE     16$             ;;BRANCH IF NO
1176  005216  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
1177  005222  001403                        BEQ     11$             ;;BRANCH IF YES
1178  005224  016677  000002  173706        MOV     2(SP),@SWR      ;;SAVE NEW SWR
1179  005232  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
1180  005236  104401  001165        14$:    TYPE    ,$C^_F          ;;ECHO <CR> AND <LF>
1181  005242  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
1182  005250  001003                        BNE     15$             ;;BRANCH IF NOT
1183  005252  012777  000100  173664        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
1184  005260  000002                15$:    RTI                     ;;RETURN
1185  005262  004737  006242        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
1186  005266  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
1187  005272  002420                        BLT     18$             ;;BRANCH IF YES
1188  005274  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
1189  005300  003015                        BGT     18$             ;;BRANCH IF YES
1190  005302  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
1191  005306  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
1192  005312  001403                        BEQ     17$             ;;BRANCH IF YES
1193  005314  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
1194  005316  006316                        ASL     (SP)            ;;    CHAR OVER TO MAKE
1195  005320  006316                        ASL     (SP)            ;;    ROOM FOR NEW ONE.
1196  005322  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
1197  005326  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
1198  005332  000707                        BR      7$              ;;GET THE NEXT ONE
1199  005334  104401  001164        18$:    TYPE    ,$QUES          ;;TYPE ?(CR)(LF)
```

```
1200  005340  000720                        BR      20$             ;;SIMULATE CONTROL-U
1201                                .DSABL  LSB
1202
1203
1204                                ;;*****************************************************************
1205                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
1206                                ;*CALL:
1207                                ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
1208                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
1209                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
1210                                ;
1211
1212  005342  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
1213  005344  016666  000004  000002        MOV     4(SP),2(SP)     ;;THE PS
1214  005352  005066  000004                CLR     4(SP)           ;;GET READY FOR A CHARACTER
1215  005356  005046                        CLR     -(SP)           ;;PUT NEW PS ON STACK
1216  005360  012746  005366                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1217  005364  000002                        RTI                     ;;POP NEW PC AND PS
1218  005366                        64$:
1219  005366  005737  004566        1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
1220  005372  001775                        BEQ     1$
1221  005374  005337  004566                DEC     $TKCNT          ;;DECREMENT THE COUNTER
1222  005400  117766  17716  000004        MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
1223  005406  005237  00452                 INC     $TKQOUT         ;;UPDATE THE POINTER
1224  005412  023727  004572  004575        CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
1225  005420  001003                        BNE     2$              ;;BRANCH IF NO
1226  005422  012737  004574  004572        MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
1227  005430  000002                2$:     RTI                     ;;RETURN
1228                                ;;*****************************************************************
1229                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
1230                                ;*CALL:
1231                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
1232                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
1233                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
1234
1235  005432  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
1236  005434  005046                        CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
1237  005436  012703  005666        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
1238  005442  022703  005675        2$:     CMP     #$TTYIN+7,R3    ;;BUFFER FULL?
1239  005446  101456                        BLOS    4$              ;;BR IF YES
1240  005450  104407                        RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
1241  005452  112613                        MOVB    (SP)+,(R3)      ;;GET CHARACTER
1242  005454  122713  000177        10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
1243  005460  001022                        BNE     5$              ;;BR IF NO
1244  005462  005716                        TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
1245  005464  001007                        BNE     6$              ;;BR IF NO
1246  005466  112737  000134  005664        MOVB    #'\,9$          ;;TYPE A BACK SLASH
1247  005474  104401  005664                TYPE    ,9$
1248  005500  012716  177777                MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
1249  005504  005303                6$:     DEC     R3              ;;BACKUP BY ONE
1250  005506  020327  005666                CMP     R3,#$TTYIN      ;;STACK EMPTY?
1251  005512  103434                        BLO     4$              ;;BR IF YES
1252  005514  111337  005664                MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
1253  005520  104401  005664                TYPE    ,9$             ;;GO TYPE
1254  005524  000746                        BR      2$              ;;GO READ ANOTHER CHAR.
1255  005526  005716                5$:     TST     (SP)            ;;RUBOUT KEY SET?
```

```
1256  005530  001406                        BEQ     7$              ;;BR IF NO
1257  005532  112737  000134  005664        MOVB    #'\,9$          ;;TYPE A BACK SLASH
1258  005540  104401  005664                TYPE    9$
1259  005544  005016                        CLR     (SP)            ;;CLEAR THE RUBOUT KEY
1260  005546  122713  000025        7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
1261  005552  001003                        BNE     8$              ;;BR IF NO
1262  005554  104401  005675                TYPE    $CNTLU          ;;TYPE A CONTROL "U"
1263  005560  000726                        BR      1$              ;;GO START OVER
1264  005562  122713  000022        8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "↑R"?
1265  005566  001011                        BNE     3$              ;;BRANCH IF NO
1266  005570  105013                        CLRB    (R3)            ;;CLEAR THE CHARACTER
1267  005572  104401  001165                TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
1268  005576  104401  005666                TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
1269  005602  000717                        BR      2$              ;;GO PICKUP ANOTHER CHACTER
1270  005604  104401  001164        4$:     TYPE    $QUES           ;;TYPE A '?'
1271  005610  000712                        BR      1$              ;;CLEAR THE BUFFER AND LOOP
1272  005612  111337  005664        3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
1273  005616  104401  005664                TYPE    9$
1274  005622  122723  000015                CMPB    #15,(R3)+       ;;CHECK FOR RETURN
1275  005626  001305                        BNE     2$              ;;LOOP IF NOT RETURN
1276  005630  105063  177777                CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
1277  005634  104401  001166                TYPE    $LF             ;;TYPE A LINE FEED
1278  005640  005726                        TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
1279  005642  012603                        MOV     (SP)+,R3        ;;RESTORE R3
1280  005644  011646                        MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
1281  005646  016666  000004  000002        MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
1282  005654  012766  005666  000004        MOV     #$TTYIN,4(SP)
1283  005662  000002                        RTI                     ;;RETURN
1284  005664     000                9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
1285  005665     000                        .BYTE   0               ;;TERMINATOR
1286  005666  000007                $TTYIN: .BLKB   7               ;;RESERVE 7 BYTES FOR TTY INPUT
1287  005675     136  006525  000012 $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
1288  005702  043536  005015     000 $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
1289  005707     015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR = /
1290  005714  036440  000040
1291  005720  020040  042516  020127 $MNEW:  .ASCIZ  /  NEW = /
1292  005726  020075     000
1293          005732                .EVEN
1294
1295                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
1296
1297                                ;;**************************************************************
1298                                ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
1299                                ;*CHANGE IT TO BINARY.
1300                                ;*THE INPUT CHARACTERS WILL BE CHECKED TO INSURED THEY ARE LEGAL
1301                                ;*OCTAL DIGITS. IF AN ILLEGAL CHARACTER IS READ A "?" WILL BE TYPED
1302                                ;*FOLLOWED BY A CARRIAGE RETURN-LINE FEED. THE COMPLETE NUMBER MUST
1303                                ;*THEN BE RETYPED. THE INPUT IS TERMINATED BY TYPING A CARRIAGE RETURN.
1304                                ;*CALL:
1305                                ;*      RDOCT                   ;;READ AN OCTAL NUMBER
1306                                ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
1307                                ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
1308
1309  005732  011646                $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
1310  005734  016666  000004  000002        MOV     4(SP),2(SP)     ;;INPUT NUMBER
1311  005742  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
```

```
1312  005744  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1313  005746  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1314  005750  104410                  1$:     RDLIN                   ;;READ AN ASCIZ LINE
1315  005752  012600                          MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
1316  005754  010037  006060                  MOV     R0,5$           ;;AND SAVE IT
1317  005760  005001                          CLR     R1              ;;CLEAR DATA WORD
1318  005762  005002                          CLR     R2
1319  005764  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
1320  005766  001420                          BEQ     3$              ;;IF ZERO GET OUT
1321  005770  122716  000060                  CMPB    #'0,(SP)        ;;MAKE SURE THIS CHARACTER
1322  005774  003026                          BGT     4$              ;;IS AN OCTAL DIGIT
1323  005776  122716  000067                  CMPB    #'7,(SP)
1324  006002  002423                          BLT     4$
1325  006004  006301                          ASL     R1              ;;*2
1326  006006  006102                          ROL     R2
1327  006010  006301                          ASL     R1              ;;*4
1328  006012  006102                          ROL     R2
1329  006014  006301                          ASL     R1              ;;*8
1330  006016  006102                          ROL     R2
1331  006020  042716  177770                  BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
1332  006024  062601                          ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
1333  006026  000756                          BR      2$              ;;LOOP
1334  006030  005726                  3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
1335  006032  010166  000012                  MOV     R1,12(SP)       ;;SAVE THE RESULT
1336  006036  010237  006070                  MOV     R2,$HIOCT
1337  006042  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
1338  006044  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1339  006046  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1340  006050  000002                          RTI                     ;;RETURN
1341  006052  005726                  4$:     TST     (SP)+           ;;CLEAN PARTIAL FROM STACK
1342  006054  105010                          CLRB    (R0)            ;;SET A TERMINATOR
1343  006056  104401                          TYPE                    ;;TYPE UP THRU THE BAD CHAR.
1344  006060  000000                  5$:     .WORD   0
1345  006062  104401  001164                  TYPE    ,$QUES          ;;"?" "CR" & "LF"
1346  006066  000730                          BR      1$              ;;TRY AGAIN
1347  006070  000000                  $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
1348
1349                                  .SBTTL  TYPE ROUTINE
1350
1351                                  ;;*****************************************************************
1352                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1353                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1354                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1355                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1356                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1357                                  ;*
1358                                  ;*CALL:
1359                                  ;*1) USING A TRAP INSTRUCTION
1360                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1361                                  ;*OR
1362                                  ;*      TYPE
1363                                  ;*      MESADR
1364                                  ;*
1365
1366  006072  105737  001157  $TYPE:   TSTB    $TPFLG          ;;IS THERE A TERMINAL?
1367  006076  100002                  BPL     1$              ;;BR IF YES
```

```
1368  006100  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
1369  006102  000407                        BR      3$              ;;LEAVE
1370  006104  010046              1$:       MOV     R0,-(SP)        ;;SAVE R0
1371  006106  017600  000002              MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
1372  006112  112046              2$:       MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1373  006114  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1374  006116  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1375  006120  012600              60$:      MOV     (SP)+,R0        ;;RESTORE R0
1376  006122  062716  000002      3$:       ADD     #2,(SP)         ;;ADJUST RETURN PC
1377  006126  000002                        RTI                     ;;RETURN
1378  006130  122716  000011      4$:       CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1379  006134  001430                        BEQ     8$
1380  006136  122716  000200              CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
1381  006142  001006                        BNE     5$
1382  006144  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
1383  006146  104402                        TYPE                    ;;TYPE A CR AND LF
1384  006150  001165                        $CRLF
1385  006152  105037  006306              CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
1386  006156  000755                        BR      2$              ;;GET NEXT CHARACTER
1387  006160  004737  006242      5$:       JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1388  006164  123726  001156      6$:       CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1389  006170  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
1390  006172  013746  001154              MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1391                                                                ;;AND THE NULL CHAR.
1392  006176  105366  000001      7$:       DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1393  006202  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1394  006204  004737  006242              JSR     PC,$TYPEC       ;;GO TYPE A NULL
1395  006210  105337  006306              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
1396  006214  000770                        BR      7$              ;;LOOP
1397
1398                                    ;HORIZONTAL TAB PROCESSOR
1399
1400  006216  112716  000040      8$:       MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1401  006222  004737  006242      9$:       JSR     PC,$TYPEC       ;;TYPE A SPACE
1402  006226  132737  000007  006306      BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1403  006234  001372                        BNE     9$              ;;TAB STOP
1404  006236  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
1405  006240  000724                        BR      2$              ;;GET NEXT CHARACTER
1406  006242  105777  172702      $TYPEC:   TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1407  006246  100375                        BPL     $TYPEC
1408  006250  116677  000002  172674      MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1409  006256  122766  000015  000002      CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
1410  006264  001003                        BNE     1$              ;;BRANCH IF NO
1411  006266  105037  006306              CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
1412  006272  000406                        BR      $TYPEX          ;;EXIT
1413  006274  122766  000012  000002  1$:       CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
1414  006302  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
1415  006304  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
1416  006306  000000              $CHARCNT: .WORD   0               ;;CHARACTER COUNT STORAGE
1417  006310  000207              $TYPEX:   RTS     PC
1418
1419
1420                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1421
1422                                    ;;**********************************************************
1423                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
```

```
                                        ;*OCTAL (ASCII) NUMBER AND TYPE IT,
                                        ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                        ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                        ;*      .BYTE   M               ;;M=1 OR 0
                                        ;*                                      ;;1=TYPE LEADING ZEROS
                                        ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                        ;*
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPON                   ;;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                        ;*      TYPOC                   ;;CALL FOR TYPEOUT

006312  017646  000000          $TYPOS: MOV     2(SP),-(SP)     ;;PICKUP THE MODE
006316  116637  000001  006535          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
006324  112637  006537                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
006330  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
006334  000406                          BR      $TYPON
006336  112737  000001  006535  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
006344  112737  000006  006537          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
006352  112737  000005  006534  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
006360  010346                          MOV     R3,-(SP)        ;;SAVE R3
006362  010446                          MOV     R4,-(SP)        ;;SAVE R4
006364  010546                          MOV     R5,-(SP)        ;;SAVE R5
006366  113704  006537                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
006372  005404                          NEG     R4
006374  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
006400  110437  006536                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
006404  113704  006535                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
006410  016505  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
006414  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
006416  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
006420  000404                          BR      3$              ;;GO DO MSB
006422  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
006424  006105                          ROL     R5
006426  006105                          ROL     R5
006430  010503                          MOV     R5,R3
006432  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
006434  105337  006536                  DECB    $OMODE          ;;TYPE THIS DIGIT?
006440  100016                          BPL     7$              ;;BR IF NO
006442  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
006446  001002                          BNE     4$              ;;TEST FOR 0
006450  005704                          TST     R4              ;;SUPPRESS THIS 0?
006452  001403                          BEQ     5$              ;;BR IF YES
006454  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
006456  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
006462  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
006466  110337  006532                  MOVB    R3,@$S          ;;SAVE FOR TYPING
```

```
1490  006472  104401  006532                  TYPE    @S              ;;GO TYPE THIS DIGIT
1491  006476  105337  006534          7$:     DECB    $OCNT           ;;COUNT BY 1
1492  006502  003347                          BGT     2$              ;;BR IF MORE TO DO
1493  006504  002402                          BLT     6$              ;;BR IF DONE
1494  006506  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
1495  006510  000744                          BR      2$              ;;GO DO THE LAST DIGIT
1496  006512  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
1497  006514  012604                          MOV     (SP)+,R4        ;;RESTORE R4
1498  006516  012603                          MOV     (SP)+,R3        ;;RESTORE R3
1499  006520  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
1500  006526  012616                          MOV     (SP)+,(SP)
1501  006530  000002                          RTI                     ;;RETURN
1502  006532     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
1503  006533     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
1504  006534     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
1505  006535     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
1506  006536  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
1507
1508                          .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES
1509
1510                          ;;*****************************************************************
1511                          ;*SAVE R0-R5
1512                          ;*CALL:
1513                          ;*      SAVREG
1514                          ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
1515                          ;*
1516                          ;*TOP---(+16)
1517                          ;* +2---(+18)
1518                          ;* +4---R5
1519                          ;* +6---R4
1520                          ;* +8---R3
1521                          ;*+10---R2
1522                          ;*+12---R1
1523                          ;*+14---R0
1524
1525  006540                  $SAVREG:
1526  006540  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
1527  006542  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1528  006544  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1529  006546  010346                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
1530  006550  010446                  MOV     R4,-(SP)        ;;PUSH R4 ON STACK
1531  006552  010546                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
1532  006554  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
1533  006560  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
1534  006564  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
1535  006570  016646  000022          MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
1536  006574  000002                  RTI
1537
1538                          ;*RESTORE R0-R5
1539                          ;*CALL:
1540                          ;*      RESREG
1541  006576                  $RESREG:
1542  006576  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
1543  006602  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
1544  006606  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
1545  006612  012666  000022          MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
```

```
1536  006616  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
1537  006620  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
1538  006622  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
1539  006624  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
1540  006626  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
1541  006630  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
1542  006632  000002                          RTI
1543
1544                                  .SBTTL  TRAP DECODER
1545
1546                                  ;;**************************************************************
1547                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1548                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1549                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1550                                  ;*GO TO THAT ROUTINE.
1551
1552  006634  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
1553  006636  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
1554  006642  005740                          TST     -(R0)           ;;BACKUP BY 2
1555  006644  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
1556  006646  006300                          ASL     R0              ;;POSITION FOR INDEXING
1557  006650  016000  006670                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
1558  006654  000200                          RTS     R0              ;;GO TO ROUTINE
1559
1560
1561                                  ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
1562
1563  006656  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
1564  006660  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
1565  006666  000002                          RTI                     ;;RESTORE THE PSW
1566
1567                                  .SBTTL  TRAP TABLE
1568
1569                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1570                                  ;*BY THE "TRAP" INSTRUCTION.
1571
1572                                  ;       ROUTINE
1573                                  ;       -------
1574  006670  006656                  $TRPAD: .WORD   $TRAP2
1575  006672  006072                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
1576  006674  006336                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
1577  006676  006312                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
1578  006700  006352                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
1579
1580  006702  005130                          $GTSWR  ;;CALL=GTSWR    TRAP+5(104405)  GET SOFT-SWR SETTING
1581
1582  006704  005040                          $CKSWR  ;;CALL=CKSWR    TRAP+6(104406)  TEST FOR CHANGE IN SOFT-SWR
1583  006706  005342                          $RDCHR  ;;CALL=RDCHR    TRAP+7(104407)  TTY TYPEIN CHARACTER ROUTINE
1584  006710  005432                          $RDLIN  ;;CALL=RDLIN    TRAP+10(104410) TTY TYPEIN STRING ROUTINE
1585  006712  005732                          $RDOCT  ;;CALL=RDOCT    TRAP+11(104411) READ AN OCTAL NUMBER FROM TTY
1586  006714  006540                          $SAVREG ;;CALL=SAVREG   TRAP+12(104412) SAVE R0-R5 ROUTINE
1587  006716  006576                          $RESREG ;;CALL=RESREG   TRAP+13(104413) RESTORE R0-R5 ROUTINE
1588
1589                                  ;THIS ROUTINE IS USED TO ENSURE THAT THE BUS ADDRESS
1590                                  ;OF THE RP11 IS SETUP TO READ THE PROPER VALUE.
1591                                  ;IT WILL ALSO READ THE ADDRESS FROM THE TTY IF
```

```
1592                                    ;REQUIRED.
1593                                    ;NOTE: THIS ROUTINE DOES NOT PROTECT R0-R4
1594
1595  006720  005737  001176  GETADR: TST     BUSADR          ;INPUT FROM TTY REQUESTED?
1596  006724  001474                  BEQ     5$              ;NO--BRANCH
1597  006726  005037  001176          CLR     BUSADR          ;YES--CLEAR THE REQUEST FLAG
1598  006732  104401  012274  1$:     TYPE    ,MRPADR         ;'RPADR='
1599  006736  013746  001264          MOV     RPADR,-(SP)     ;;SAVE RPADR FOR TYPEOUT
1600                                                          ;;RP11 ADDRESS
1601  006742  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
1602  006744     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
1603  006745     001                  .BYTE   1               ;;TYPE LEADING ZEROS
1604  006746  104401  006754          TYPE    ,65$            ;;TYPE ASCIZ STRING
1605  006752  000401                  BR      64$             ;;GET OVER THE ASCIZ
1606                          ;;65$:   .ASCIZ  @/@
1607  006756                  64$:
1608  006756  104411                  RDOCT                   ;GET NEW RP11 ADDRESS
1609  006760  012600                  MOV     (SP)+,R0        ;GET THE NEW ADDRESS
1610  006762  001402                  BEQ     2$              ;BR IF ZERO ENTRY
1611  006764  010037  001264          MOV     R0,RPADR        ;STORE THE ADDRESS
1612  006770  104401  012303  2$:     TYPE    ,MRPVEC         ;'RPVEC='
1613  006774  013746  001266          MOV     RPVEC,-(SP)     ;;SAVE RPVEC FOR TYPEOUT
1614                                                          ;;RP11 VECTOR ADDRESS
1615  007000  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
1616  007002     006                  .BYTE   6               ;;TYPE 6 DIGIT(S)
1617  007003     001                  .BYTE   1               ;;TYPE LEADING ZEROS
1618  007004  104401  007012          TYPE    ,67$            ;;TYPE ASCIZ STRING
1619  007010  000401                  BR      66$             ;;GET OVER THE ASCIZ
1620                          ;;67$:   .ASCIZ  @/@
1621  007014                  66$:
1622  007014  104411                  RDOCT                   ;READ NEW RP11 VECTOR
1623  007016  012600                  MOV     (SP)+,R0        ;GET THE NEW VECTOR
1624  007020  001402                  BEQ     3$              ;BR IF ZERO ENTRY
1625  007022  010037  001266          MOV     R0,RPVEC        ;SAVE NEW RP11 VECTOR ADDRESS
1626  007026  104401  012312  3$:     TYPE    ,MRPPRI         ;'RPPRIO='
1627  007032  013700  001272          MOV     RPPRIO,R0       ;CONVERT PRIORITY FOR TYPEOUT
1628  007036  006300                  ASL     R0
1629  007040  006300                  ASL     R0
1630  007042  006300                  ASL     R0
1631  007044  000300                  SWAB    R0              ;ALIGN FOR TYPEOUT
1632  007046  010046                  MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
1633                                                          ;;RP11 PRIORITY
1634  007050  104403                  TYPOS                   ;;GO TYPE--OCTAL ASCII
1635  007052     001                  .BYTE   1               ;;TYPE 1 DIGIT(S)
1636  007053     001                  .BYTE   1               ;;TYPE LEADING ZEROS
1637  007054  104401  007062          TYPE    ,69$            ;;TYPE ASCIZ STRING
1638  007060  000401                  BR      68$             ;;GET OVER THE ASCIZ
1639                          ;;69$:   .ASCIZ  @/@
1640  007064                  68$:
1641  007064  104411                  RDOCT                   ;GET NEW PRIORITY LEVEL
1642  007066  012600                  MOV     (SP)+,R0        ;SAVE NEW PRIORITY
1643  007070  001407                  BEQ     4$              ;BR IF ZERO ENTRY
1644  007072  006300                  ASL     R0              ;CONVERT TO PRIOIRITY VALUE
1645  007074  006300                  ASL     R0              ;CONVERT TO PRIOIRITY VALUE
1646  007076  006300                  ASL     R0              ;CONVERT TO PRIOIRITY VALUE
1647  007100  006300                  ASL     R0              ;CONVERT TO PRIOIRITY VALUE
```

```
1648  007102  006300                          ASL     R0              ;CONVERT TO PRIOIRITY VALUE
1649  007104  010037  001272                  MOV     R0,RPPRIO       ;SAVE THE VALUE
1650  007110  013777  001272  172152  4$:     MOV     RPPRIO,@RPVEC+2 ;LOAD NEW PRIORITY VALUE
1651  007116  013701  000004          5$:     MOV     ERRVEC,R1       ;SAVE THE ERROR VECTOR
1652  007122  012737  007142  000004          MOV     #6$,ERRVEC      ;SETUP FOR TRAP
1653  007130  005777  172130                  TST     @RPADR          ;CHECK FOR RP11
1654  007134  010137  000004                  MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
1655  007140  000207                          RTS     PC              ;RETURN
1656  007142  010137  000004          6$:     MOV     R1,ERRVEC       ;RESTORE ERROR VECTOR
1657  007146  022626                          CMP     (SP)+,(SP)+     ;CLEAN OFF THE STACK
1658  007150  104401  007156                  TYPE    ,71$            ;TYPE ASCIZ STRING
1659  007154  000424                          BR      70$             ;GET OVER THE ASCIZ
1660                                  71$:    .ASCIZ  <15><12><12>/RP11 DIDN'T RESPOND TO ADDRESSING/<15><12>
1661  007226                          70$:
1662  007226  000641                          BR      1$              ;GO ASK FOR ADDRESS
1663
1664
1665                                  ;*THIS ROUTINE CLEARS THE RP11 AND DETERMINES WHICH DRIVES ARE AVAILABLE.
1666                                  ;*THE TABLE 'DRVSTA' IS LOADED TO REFLECT THE SYSTEM STATUS.
1667
1668  007230  104412                  RPINIT: SAVREG                  ;SAVE R0-R5
1669  007232  013701  001266                  MOV     RPVEC,R1        ;VECTOR ADDRESS
1670  007236  005021                          CLR     (R1)+           ;SET INTERRUPT ADDRESS TO ZERO
1671  007240  013711  001272                  MOV     RPPRIO,(R1)     ;RP11 PRIORITY
1672  007244  005037  001172                  CLR     DRVTYP          ;CLEAR DRIVE TYPE STORAGE
1673  007250  005037  001274                  CLR     DRVSTA          ;SET DRIVE STATUS TO OFFLINE
1674  007254  005037  001276                  CLR     DRVSTA+2        ;SET DRIVE STATUS TO OFFLINE
1675  007260  005037  001300                  CLR     DRVSTA+4        ;SET DRIVE STATUS TO OFFLINE
1676  007264  005037  001302                  CLR     DRVSTA+6        ;SET DRIVE STATUS TO OFFLINE
1677  007270  013704  001264                  MOV     RPADR,R4        ;PUT RP11 ADDRESS INTO R4
1678  007274  012764  000001  000004          MOV     #1,RPCS(R4)     ;CLEAR THE CONTROLLER
1679  007302  012764  000001  000004          MOV     #1,RPCS(R4)     ;CLEAR THE CONTROLLER
1680  007310  005001                          CLR     R1              ;TABLE POINTER
1681  007312  012702  000001                  MOV     #1,R2           ;DRIVE COUNTER
1682  007316  110164  000005          1$:     MOVB    R1,RPCS+1(R4)   ;SELECT DRIVE
1683  007322  032764  001000  000000          BIT     #BIT09,RPDS(R4)  ;SEE IF DRIVE UNSAFE
1684  007330  001404                          BEQ     2$              ;BR IF NOT UNSAFE
1685  007332  112761  177777  001274          MOVB    #-1,DRVSTA(R1)  ;SET INDICATOR TO 'UNSAFE'
1686  007340  000417                          BR      4$              ;CHECK ON DRIVE TYPE
1687  007342  032764  040000  000000  2$:     BIT     #BIT14,RPDS(R4)  ;IS DRIVE ONLINE ?
1688  007350  001421                          BEQ     5$              ;BR IF NOT
1689  007352  032764  004000  000000          BIT     #BIT11,RPDS(R4) ;SEEK INCOMPLETE
1690  007360  001004                          BNE     3$              ;BR IF SET
1691  007362  032764  100000  000000          BIT     #BIT15,RPDS(R4)  ;IS DRIVE READY ?
1692  007370  001411                          BEQ     5$              ;BR IF NOT
1693  007372  112761  000001  001274  3$:     MOVB    #1,DRVSTA(R1)   ;SET DRIVE INDICATOR TO 'ONLINE'
1694  007400  032764  020000  000000  4$:     BIT     #BIT13,RPDS(R4)  ;IS THE DRIVE AN RP03 ?
1695  007406  001402                          BEQ     5$              ;BR IF NOT
1696  007410  150237  001172                  BISB    R2,DRVTYP       ;SET RP03 INDICATOR
1697  007414  005201                  5$:     INC     R1              ;INCREMENT THE TABLE INDEX
1698  007416  106302                          ASLB    R2              ;SHIFT THE DRIVE COUNTER
1699  007420  103336                          BCC     1$              ;BR IF NOT FINISHED
1700  007422  104413                          RESREG                  ;RESTORE R0-R5
1701  007424  000207                          RTS     PC              ;RETURN
1702
1703                                  ;ROUTINE TO GET THE DRIVE NUMBER
```

```
1704
1705  007426  104401  011234          UNIQ:   TYPE    ,UMES
1706  007432  004737  010122                  JSR     PC,TTIO         ;READ UNIT #
1707  007436  000773                          BR      UNIQ            ;ERROR RECYCLE
1708  007440  020027  000007                  CMP     R0,#7           ;TOO BIG
1709  007444  101370                          BHI     UNIQ            ;YES
1710  007446  010037  001222                  MOV     R0,UNIT         ;SAVE IT
1711  007452  012700  000001                  MOV     #1,R0           ;DETERMINE ATTENTION BIT
1712  007456  005737  001222                  TST     UNIT            ;NO  ROTATION
1713  007462  001406                          BEQ     UNS1            ;IF 0
1714  007464  013737  001222  007476          MOV     UNIT,DS1        ;NUMBER OF ROTATES
1715  007472  004737  010250                  JSR     PC,RTL          ;TO DETERMINE ATTENTION BIT
1716  007476  000001                  DS1:    .WORD   1               ;IS PUT HERE
1717  007500  010037  001224          UNS1:   MOV     R0,ATNB         ;SAVE BIT
1718  007504  113764  001222  000005  1$:     MOVB    UNIT,RPCS+1(R4) ;SELECT UNIT
1719  007512  032764  100000  000000          BIT     #BIT15,RPDS(R4) ;IS IT READY?
1720  007520  001413                          BEQ     3$              ;NO
1721  007522  012737  000312  001170          MOV     #202.,MAXCYL    ;ASSUME AN RP02
1722  007530  033737  001224  001172          BIT     ATNB,DRVTYP     ;SEE WHICH
1723  007536  001403                          BEQ     2$              ;BR IF RP02
1724  007540  012737  000625  001170          MOV     #405.,MAXCYL    ;SETUP FOR RP03
1725  007546  000207                  2$:     RTS     PC              ;EXIT
1726  007550  104034                  3$:     ERROR   DS!ER!CS        ;SELECTED UNIT READY NOT SET
1727  007552  000754                          BR      1$              ;LOOP UNTIL FIXED
1728
1729                                  ;TYPE SPACES.  NUMBER OF SPACES IN IN 'TTG'
1730
1731  007554  104401  012050          SPACE:  TYPE    ,S1
1732  007560  005337  001250                  DEC     TTG             ;PRINT NUMBER OF SPACES
1733  007564  001373                          BNE     SPACE           ;IN TTG. THEN
1734  007566  000207                          RTS     PC              ;EXIT
1735
1736                                  ;GET PARAMETERS
1737
1738  007570  004737  010122          PARIN:  JSR     PC,TTIO         ;READ SOME NUMBERS
1739  007574  000401                          BR      .+4             ;THIS NOW THE GOOD RETURN
1740  007576  000207                          RTS     PC              ;'CR' IS ILLEGAL FOR THIS SEQUENCE
1741  007600  020027  000040                  CMP     R0,#40          ;IS IT A SPACE ?
1742  007604  001015                          BNE     1$              ;NO - EXIT
1743  007606  023705  001214                  CMP     WORK1,R5        ;TOO BIG ?
1744  007612  101012                          BHI     1$              ;YES - EXIT
1745  007614  005237  001250                  INC     TTG             ;TTG=# OF SPACES
1746  007620  023727  001250  000005          CMP     TTG,#5          ;TOO MANY
1747  007626  101004                          BHI     1$              ;YES
1748  007630  004737  007554                  JSR     PC,SPACE        ;JUSTIFY NEXT ENTRY
1749  007634  062716  000002                  ADD     #2,(SP)         ;SKIP FOR GOOD RETURN
1750  007640  000207                  1$:     RTS     PC              ;EXIT
1751
1752                                  ;ROUTINE TO DO A HOME SEEK
1753
1754  007642  004737  011216          HOME:   JSR     PC,CLRP         ;CLEAR THE CONTROLLER
1755  007646  113764  001222  000005          MOVB    UNIT,RPCS+1(R4)  ;SELECT THE DRIVE
1756  007654  112764  000015  000004          MOVB    #15,RPCS(R4)    ;HOME SEEK
1757  007662  012746  000005                  MOV     #5,-(SP)        ;WAIT FOR
1758  007666  005316                  1$:     DEC     (SP)            ;SEEK TO
1759  007670  001376                          BNE     1$              ;START
```

```
 1760  007672  005726                          TST     (SP)+           ;RESTORE THE STACK POINTER
 1761  007674  032764  002000  000000          BIT     #BIT10,RPDS(R4) ;IS SEEK UNDERWAY?
 1762  007702  001002                          BNE     2$              ;YES
 1763  007704  104034                          ERROR   ER!DS!CS        ;SEEK NOT UNDERWAY
 1764  007706  000755                          BR      HOME            ;FIX IT
 1765  007710  033764  001224  000000  2$:     BIT     ATNB,RPDS(R4)   ;WAIT FOR
 1766  007716  001774                          BEQ     2$              ;ATTENTION TO SET
 1767  007720  032764  004000  000000          BIT     #BIT11,RPDS(R4) ;SEEK INCOMPLETE?
 1768  007726  001402                          BEQ     3$              ;NO
 1769  007730  104034                          ERROR   ER!DS!CS        ;DEVICE STATUS ERROR
 1770  007732  000743                          BR      HOME            ;LOOP
 1771  007734  132764  000023  000001  3$:     BITB    #23,RPDS+1(R4)  ;ANY ERRORS?
 1772  007742  001402                          BEQ     4$              ;NO
 1773  007744  104034                          ERROR   ER!CS!DS        ;DEVICE STATUS ERROR
 1774  007746  000735                          BR      HOME            ;LOOP
 1775  007750  112764  000377  000000  4$:     MOVB    #377,RPDS(R4)   ;CLEAR ATTN BITS
 1776  007756  000207                          RTS     PC              ;EXIT
 1777
 1778                                  ;CHECK THE HEADER
 1779
 1780  007760  062703  000002          CHKAD:  ADD     #2,R3           ;SKIP 1ST WORD
 1781  007764  012337  001236                  MOV     (R3)+,HEDR      ;FETCH HEAD AND
 1782  007770  013737  001236  001234          MOV     HEDR,CYLR       ;CYLINDER READ
 1783  007776  006037  001236                  ROR     HEDR            ;JUSTIFY HEAD
 1784  010002  006037  001234                  ROR     CYLR
 1785  010006  006037  001234                  ROR     CYLR
 1786  010012  006037  001234                  ROR     CYLR
 1787  010016  006037  001234                  ROR     CYLR
 1788  010022  006037  001234                  ROR     CYLR
 1789  010026  006037  001234                  ROR     CYLR
 1790  010032  012337  001240                  MOV     (R3)+,SECR      ;FETCH SECTOR READ
 1791  010036  042737  177760  001240          BIC     #^C17,SECR
 1792  010044  042737  177740  001236          BIC     #^C37,HEDR
 1793  010052  042737  177000  001234          BIC     #^C777,CYLR
 1794  010060  023737  001226  001234          CMP     CYL,CYLR        ;IS CYL# OK?
 1795  010066  001010                          BNE     1$              ;NO
 1796  010070  023737  001230  001236          CMP     HED,HEDR        ;IS TRACK# OK
 1797  010076  001004                          BNE     1$              ;NO
 1798  010100  023737  001232  001240          CMP     SEC,SECR        ;IS SECTOR # OK?
 1799  010106  001402                          BEQ     2$              ;YES
 1800  010110  104100                  1$:     ERROR   HD              ;HEADER COMPARE ERROR
 1801  010112  000402                          BR      3$              ;EXIT
 1802  010114  062716  000002          2$:     ADD     #2,(SP)         ;INCREMENT RETURN
 1803  010120  000207                  3$:     RTS     PC              ;RETURN
 1804
 1805                                  ;READ AND CHECK OCTAL DATA FROM THE KEYBOARD
 1806
 1807  010122  012737  000005  001250  TTIO:   MOV     #5,TTG          ;INPUT UP TO 5 OCTAL DIGITS
 1808  010130  005037  001214                  CLR     WORK1           ;CLEAR WORK REGISTER
 1809  010134  004737  010242          1$:     JSR     PC,RIN          ;READ TTY CHARACTER INTO R0
 1810  010140  020027  000015                  CMP     R0,#15          ;IS IT CR?
 1811  010144  001427                          BEQ     2$              ;YES
 1812  010146  004737  011154                  JSR     PC,TTO          ;ECHO
 1813  010152  020027  000067                  CMP     R0,#67          ;TOO BIG?
 1814  010156  101030                          BHI     3$              ;YES
 1815  010160  020027  000057                  CMP     R0,#57          ;TOO SMALL?
```

```
1816  010164  101425                          BLOS    3$              ;YES
1817  010166  005337  001250                  DEC     TTG             ;COUNT DIGIT
1818  010172  100422                          BMI     3$              ;TOO MANY?
1819  010174  042700  177770                  BIC     #^C7,R0         ;NO - MASK IT
1820  010200  000241                          CLC                     ;AND
1821  010202  006137  001214                  ROL     WORK1           ;SHIFT
1822  010206  006137  001214                  ROL     WORK1           ;WORK REGISTER
1823  010212  006137  001214                  ROL     WORK1           ;3 PLACES
1824  010216  060037  001214                  ADD     R0,WORK1        ;ADD NEW NUMBER
1825  010222  000744                          BR      1$              ;GET NEXT ONE
1826  010224  104401  001165          2$:     TYPE    ,$CRLF          ;CR-LF
1827  010230  062716  000002                  ADD     #2,(SP)         ;SKIP RETURN
1828  010234  013700  001214                  MOV     WORK1,R0
1829  010240  000207                  3$:     RTS     PC
1830
1831                                  ;READ A CHARACTER FROM THE KEYBOARD AND RETURN WITH
1832                                  ;       THE CHARACTER IN R0.
1833
1834  010242  104407                  RIN:    RDCHR                   ;READ THE KEYBOARD
1835  010244  012600                          MOV     (SP)+,R0        ;PUT THE CHARACTER IN R0
1836  010246  000207                          RTS     PC              ;RETURN
1837
1838                                  ;GENERATE THE ATTENTION BIT FOR THE DRIVE NUMBER
1839                                  ;       R0 IS PRELOADED WITH BIT 0, RETURN WITH ATTN BIT IN R0.
1840
1841  010250  017637  000000  001256  RTL:    MOV     @(SP),ROTOG     ;GET THE SHIFT COUNT
1842  010256  062716  000002                  ADD     #2,(SP)         ;CORRECT RETURN ADDRESS
1843  010262  005337  001256          1$:     DEC     ROTOG           ;DECREMENT THE COUNT
1844  010266  100402                          BMI     2$              ;EXIT WHEN DONE
1845  010270  006300                          ASL     R0              ;SHIFT THE BIT
1846  010272  000773                          BR      1$              ;CONTINUE
1847  010274  000207                  2$:     RTS     PC              ;EXIT
1848
1849                                  ;PROCESS THE ERROR CALL
1850
1851  010276  032777  002000  170634  $ERROR: BIT     #SW10,@SWR              ;RING THE BELL?
1852  010304  001402                          BEQ     1$              ;BRANCH IF NO
1853  010306  104401  001160                  TYPE    ,$BELL
1854  010312  032777  020000  170620  1$:     BIT     #SW13,@SWR      ;TYPE ERROR MESSAGE ?
1855  010320  001064                          BNE     2$              ;EXIT IF SWITCH 13 IS SET
1856  010322  104412                          SAVREG                  ;SAVE R0-R5
1857  010324  104401  012052                  TYPE    ,ERMS1
1858  010330  011605                          MOV     (SP),R5         ;FETCH SAVED PC
1859  010332  162705  000002                  SUB     #2,R5           ;FIND ADDRESS OF ERROR CALL
1860  010336  010546                          MOV     R5,-(SP)        ;PUT ADDRESS ON THE STACK
1861  010340  104402                          TYPOC                   ;TYPE THE CALLING PC
1862  010342  104401  012231                  TYPE    ,STMS           ;TYPE 'PS='
1863  010346  016600  000002                  MOV     2(SP),R0        ;STATUS AT TIME OF ERROR CALL
1864  010352  010046                          MOV     R0,-(SP)        ;PUT IT ON THE STACK
1865  010354  104402                          TYPOC                   ;TYPE IT
1866  010356  032715  000020                  BIT     #BIT04,(R5)     ;RPCS?
1867  010362  001402                          BEQ     .+6             ;NO
1868  010364  004737  010542                  JSR     PC,CSTYPE       ;YES
1869  010370  032715  000010                  BIT     #BIT03,(R5)     ;RPER?
1870  010374  001402                          BEQ     .+6             ;NO
1871  010376  004737  010572                  JSR     PC,ERTYPE       ;YES
```

```
1872  010402  032715  000004                  BIT     #BIT02,(R5)     ;RPDS?
1873  010406  001402                          BEQ     .+6             ;NO
1874  010410  004737  010606                  JSR     PC,DSTYPE       ;YES
1875  010414  032715  000040                  BIT     #BIT05,(R5)     ;RPDA & RPCA?
1876  010420  001402                          BEQ     .+6             ;NO
1877  010422  004737  010622                  JSR     PC,ADTYPE       ;YES
1878  010426  032715  000100                  BIT     #BIT06,(R5)     ;HEADER MESSAGE?
1879  010432  001402                          BEQ     .+6             ;NO
1880  010434  004737  011026                  JSR     PC,HDTYPE       ;YES
1881  010440  032715  000001                  BIT     #BIT00,(R5)     ;BIT 0 FOR GOOD/BAD
1882  010444  001402                          BEQ     .+6             ;SKIP IF=0
1883  010446  004737  010514                  JSR     PC,GBTYPE       ;GOOD/BAD
1884  010452  032715  000002                  BIT     #BIT01,(R5)     ;BIT 1 FOR DATA
1885  010456  001402                          BEQ     .+6             ;SKIP IF=0
1886  010460  004737  010556                  JSR     PC,DATYPE       ;DATA
1887  010464  104413                          RESREG                  ;RESTORE R0-R5
1888  010466  104401  001165                  TYPE    ,$CRLF          ;CR-LF
1889  010472  105764  000004          2$:     TSTB    RPCS(R4)        ;WAIT FOR DONE
1890  010476  100005                          BPL     3$
1891  010500  032777  100000  170432          BIT     #SW15,@SWR      ;HALT ON ERROR?
1892  010506  001401                          BEQ     3$
1893  010510  000000                          HALT                    ;ERROR HALT. CONTINUE
1894  010512  000002                  3$:     RTI                     ;EXIT ERROR ROUTINE
1895
1896                                  ;TYPE GOOD/BAD DATA
1897
1898  010514  104401  012212          GBTYPE: TYPE    ,GDMS           ;'GOOD'
1899  010520  013746  001200                  MOV     GOOD,-(SP)      ;PUT IT ON THE STACK
1900  010524  104402                          TYPOC                   ;TYPE IT
1901  010526  104401  012223                  TYPE    ,BDMS           ;TYPE 'BAD'
1902  010532  013746  001202                  MOV     BAD,-(SP)       ;PUT IT ON THE STACK
1903  010536  104402                          TYPOC                   ;TYPE IT
1904  010540  000207                          RTS     PC              ;EXIT
1905
1906                                  ;TYPE THE CONTENTS OF 'RPCS'
1907
1908  010542  104401  012127          CSTYPE: TYPE    ,CSMS           ;'STATUS'
1909  010546  016446  000004                  MOV     RPCS(R4),-(SP)  ;PUT RPCS ON THE STACK
1910  010552  104402                          TYPOC                   ;TYPE IT
1911  010554  000207                          RTS     PC              ;EXIT
1912
1913                                  ;TYPE THE DATA
1914
1915  010556  104401  012240          DATYPE: TYPE    ,DAMS           ;'DATA='
1916  010562  013746  001204                  MOV     DATA,-(SP)      ;PUT DATA ON THE STACK
1917  010566  104402                          TYPOC                   ;TYPE IT
1918  010570  000207                          RTS     PC              ;EXIT
1919
1920                                  ;TYPE THE ERROR REGISTER (RPER)
1921
1922  010572  104401  012140          ERTYPE: TYPE    ,ERMS           ;'RPER='
1923  010576  016446  000002                  MOV     RPER(R4),-(SP)  ;PUT RPER ON THE STACK
1924  010602  104402                          TYPOC                   ;TYPE IT
1925  010604  000207                          RTS     PC              ;EXIT
1926
1927                                  ;TYPE THE DRIVE STATUS REGISTR (RPDS)
```

```
1928
1929  010606  104401  012151          DSTYPE: TYPE    ,DSMS               ;'RPDS:'
1930  010612  016446  000000                  MOV     RPDS(R4),-(SP)      ;PUT RPDS ON THE STACK
1931  010616  104402                         TYPOC                       ;TYPE IT
1932  010620  000207                         RTS     PC                  ;EXIT
1933
1934                                  ;TYPE THE DISK ADDRESS
1935
1936  010622  032764  000001  000002  ADTYPE: BIT     #BIT00,RPER(R4)     ;ADDRESS ERROR?
1937  010630  001407                         BEQ     1$                  ;BRANCH IF NO
1938  010632  016437  000012  001220         MOV     RPCA(R4),WORK3      ;GET THE CLYINDER ADDRESS
1939  010640  016437  000014  001216         MOV     RPDA(R4),WORK2      ;GET HEAD AND SECTOR ADDRESS
1940  010646  000437                         BR      ADT1
1941  010650  016437  000012  001220  1$:    MOV     RPCA(R4),WORK3
1942  010656  016437  000014  001216         MOV     RPDA(R4),WORK2
1943  010664  042737  160360  001216         BIC     #160360,WORK2
1944  010672  032737  000017  001216         BIT     #17,WORK2
1945  010700  001403                         BEQ     DECTK
1946  010702  005337  001216                 DEC     WORK2
1947  010706  000417                         BR      ADT1
1948  010710  132737  000037  001217  DECTK:  BITB    #37,WORK2+1
1949  010716  001406                         BEQ     DECCY
1950  010720  105337  001217                 DECB    WORK2+1
1951  010724  052737  000011  001216         BIS     #11,WORK2
1952  010732  000405                         BR      ADT1
1953  010734  012737  011411  001216  DECCY:  MOV     #11411,WORK2
1954  010742  005337  001220                 DEC     WORK3
1955  010746  104401  012162          ADT1:   TYPE    ,CYMS               ;'CAR='
1956  010752  013746  001220                 MOV     WORK3,-(SP)         ;MPUT CAR ON THE STACK
1957  010756  042716  177400                 BIC     #^C377,(SP)
1958  010762  104402                         TYPOC                       ;TYPE IT
1959  010764  104401  012172                 TYPE    ,TAMS               ;'TAR='
1960  010770  005046                         CLR     -(SP)               ;CLEAR THE STACK
1961  010772  113716  001217                 MOVB    WORK2+1,(SP)        ;PUT THE TRACK ON THE STACK
1962  010776  042716  177740                 BIC     #^C37,(SP)
1963  011002  104402                         TYPOC                       ;TYPE IT
1964  011004  104401  012202                 TYPE    ,SEMS               ;'SAR='
1965  011010  005046                         CLR     -(SP)               ;CLEAR THE STACK
1966  011012  113716  001216                 MOVB    WORK2,(SP)          ;PUT THE SECTOR ADDRESS ON THE STACK
1967  011016  042716  177760                 BIC     #^C17,(SP)
1968  011022  104402                         TYPOC                       ;TYPE IT
1969  011024  000207                         RTS     PC                  ;EXIT
1970
1971
1972
1973  011026  104401  012010          HDTYPE: TYPE    ,HDMS
1974  011032  104401  012212                 TYPE    ,GDMS
1975  011036  104401  012047                 TYPE    ,S2
1976  011042  013746  001226                 MOV     CYL,-(SP)           ;;SAVE CYL FOR TYPEOUT
1977  011046  104402                         TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1978  011050  104401  012050                 TYPE    ,S1
1979  011054  013746  001230                 MOV     HED,-(SP)           ;;SAVE HED FOR TYPEOUT
1980  011060  104402                         TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1981  011062  104401  012050                 TYPE    ,S1
1982  011066  013746  001232                 MOV     SEC,-(SP)           ;;SAVE SEC FOR TYPEOUT
1983  011072  104402                         TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
1984  011074  104401  001165                  TYPE    ,$CRLF
1985  011100  104401  012050                  TYPE    ,S1
1986  011104  104401  012223                  TYPE    ,BOMS
1987  011110  104401  012047                  TYPE    ,S2
1988  011114  013746  001234                  MOV     CYLR,-(SP)      ;;SAVE CYLR FOR TYPEOUT
1989  011120  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1990  011122  104401  012050                  TYPE    ,S1
1991  011126  013746  001236                  MOV     HEDR,-(SP)      ;;SAVE HEDR FOR TYPEOUT
1992  011132  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1993  011134  104401  012050                  TYPE    ,S1
1994  011140  013746  001240                  MOV     SECR,-(SP)      ;;SAVE SECR FOR TYPEOUT
1995  011144  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
1996  011146  104401  001165                  TYPE    ,$CRLF
1997  011152  000207                          RTS     PC
1998
1999                                  ;TYPE THE CHARACTER IN R0
2000
2001  011154  022700  000015          TTO:    CMP     #15,R0          ;PRINT A 'CR' ?
2002  011160  001405                          BEQ     1$              ;BR IF 'CR'
2003  011162  110037  011202                  MOVB    R0,3$           ;SETUP 'ASCIZ' FIELD
2004  011166  104401  011202                  TYPE    ,3$             ;TYPE IT
2005  011172  000402                          BR      2$              ;EXIT
2006  011174  104401  001165          1$:     TYPE    ,$CRLF          ;TYPE A CR-LF
2007  011200  000207                  2$:     RTS     PC              ;RETURN
2008  011202     000     000          3$:     .BYTE   0,0             ;CHARACTER STORAGE AND TERMINATOR
2009
2010                                  ;TYPE THE MESSAGE WHOSE ADDRESS IS IN R0
2011
2012  011204  010037  011212          TYPOUT: MOV     R0,1$           ;STORE THE ADDRESS
2013  011210  104401                          TYPE                    ;TYPE THE MESSAGE
2014  011212  000000                  1$:     .WORD   0               ;MESSAGE ADDRESS
2015  011214  000207                          RTS     PC              ;RETURN
2016
2017                                  ;CLEAR THE CONTROLLER
2018
2019  011216  012764  000001  000004  CLRP:   MOV     #1,RPCS(R4)     ;CLEAR THE CONTROLLER
2020  011224  012764  000001  000004          MOV     #1,RPCS(R4)     ;CLEAR THE CONTROLLER AGAIN
2021  011232  000207                          RTS     PC              ;RETURN
2022
2023                                  .SBTTL  TELETYPE MESSAGES
2024
2025  011234  006015  042012  044522  UMES:   .ASCIZ  <15><12><12>/DRIVE #: /
2026  011242  042526  021440  020072
2027  011250     000
2028  011251     015  005012  047111  SECMES: .ASCIZ  <15><12><12>/INPUT THE SECTOR NUMBERS (0-9) IN THE ORDER DESIRED./
2029  011256  052520  020124  044124
2030  011264  020105  042523  052103
2031  011272  051117  047040  046525
2032  011300  042502  051522  024040
2033  011306  026460  024471  044440
2034  011314  020116  044124  020105
2035  011322  051117  042504  020122
2036  011330  042504  044523  042522
2037  011336  027104     000
2038  011341     015  030012  020072  NO:     .ASCIZ  <15><12>/0: /
2039  011346     000
```

```
  2040  011347     015  030412  020072  N1:     .ASCIZ  <15><12>/1: /
  2041  011354     000
  2042  011355     015  031012  020072  N2:     .ASCIZ  <15><12>/2: /
  2043  011362     000
  2044  011363     015  031412  020072  N3:     .ASCIZ  <15><12>/3: /
  2045  011370     000
  2046  011371     015  032012  020072  N4:     .ASCIZ  <15><12>/4: /
  2047  011376     000
  2048  011377     015  032412  020072  N5:     .ASCIZ  <15><12>/5: /
  2049  011404     000
  2050  011405     015  033012  020072  N6:     .ASCIZ  <15><12>/6: /
  2051  011412     000
  2052  011413     015  033412  020072  N7:     .ASCIZ  <15><12>/7: /
  2053  011420     000
  2054  011421     015  034012  020072  N8:     .ASCIZ  <15><12>/8: /
  2055  011426     000
  2056  011427     015  034412  020072  N9:     .ASCIZ  <15><12>/9: /
  2057  011434     000
  2058  011435     015  005012  042523  SWMES:  .ASCII  <15><12><12>/SET THE FORMAT ENABLE SWITCH./
  2059  011442  020124  044124  020105
  2060  011450  047506  046522  052101
  2061  011456  042440  040516  046102
  2062  011464  020105  053523  052111
  2063  011472  044103     056
  2064  011475     015  051412  052105          .ASCII  <15><12>/SET THE RP11 WRITE ENABLE SWITCH./
  2065  011502  052040  042510  051040
  2066  011510  030520  020061  051127
  2067  011516  052111  020105  047105
  2068  011524  041101  042514  051440
  2069  011532  044527  041524  027110
  2070  011540  005015  042523  020124          .ASCII  <15><12>/SET THE SELECTED UNIT WRITE ENABLE SWITCH./
  2071  011546  044124  020105  042523
  2072  011554  042514  052103  043105
  2073  011562  052440  044516  020124
  2074  011570  051127  052111  020105
  2075  011576  047105  041101  042514
  2076  011604  051440  044527  041524
  2077  011612  027110
  2078  011614  005015  052123  044522          .ASCIZ  <15><12>/STRIKE ANY TELETYPE KEY WHEN READY./<15><12>
  2079  011622  042513  040440  054516
  2080  011630  052040  046105  052105
  2081  011636  050131  020105  042513
  2082  011644  020131  044127  047105
  2083  011652  051040  040505  054504
  2084  011660  006456  000012
  2085  011664  005015  051012  051505  NSWMES: .ASCII  <15><12><12>/RESET THE FORMAT ENABLE SWITCH TO NORMAL./
  2086  011672  052105  052040  042510
  2087  011700  043040  051117  040515
  2088  011706  020124  047105  041101
  2089  011714  042514  051440  044527
  2090  011722  041524  020110  047524
  2091  011730  047040  051117  040515
  2092  011736  027114
```

```
2093  011740  005015  052123  044522          .ASCIZ  <15><12>/STRIKE ANY TELETYPE KEY WHEN READY./<15><12>
2094  011746  042513  040440  054516
2095  011754  052040  046105  052105
2096  011762  050131  020105  042513
2097  011770  020131  044127  047105
2098  011776  051040  040505  054504
2099  012004  006456  000012
2100  012010  005015  020040  020040  HDMS:   .ASCIZ  <15><12>/     CYLINDER  TRACK SECTOR/
2101  012016  020040  054503  044514
2102  012024  042116  051105  020040
2103  012032  051124  041501  020113
2104  012040  042523  052103  051117
2105  012046     000
2106  012047     040                  S2:     .ASCII  / /
2107  012050  000040                  S1:     .ASCIZ  / /
2108  012052  005015  042412  051122  ERMS1:  .ASCIZ  <15><12><12>/ERROR AT ADDRESS /
2109  012060  051117  040440  020124
2110  012066  042101  051104  051505
2111  012074  020123     000
2112  012077     015  047412  042114  OLDMS:  .ASCIZ  <15><12>/OLD:     /
2113  012104  020072  020040  020040
2114  012112     000
2115  012113     015  047012  053505  NEWMS:  .ASCIZ  <15><12>/NEW:     /
2116  012120  020072  020040  020040
2117  012126     000
2118  012127     015  051012  041520  CSMS:   .ASCIZ  <15><12>/RPCS= /
2119  012134  036523  000040
2120  012140  005015  050122  051105  ERMS:   .ASCIZ  <15><12>/RPER= /
2121  012146  020075     000
2122  012151     015  051012  042120  DSMS:   .ASCIZ  <15><12>/RPDS= /
2123  012156  036523  000040
2124  012162  005015  040503  036522  CYMS:   .ASCIZ  <15><12>/CAR= /
2125  012170  000040
2126  012172  020040  040524  036522  TAMS:   .ASCIZ  /  TAR= /
2127  012200  000040
2128  012202  020040  040523  036522  SEMS:   .ASCIZ  /  SAR= /
2129  012210  000040
2130  012212  005015  047507  042117  GDMS:   .ASCIZ  <15><12>/GOOD= /
2131  012220  020075     000
2132  012223     102  042101  020075  BDMS:   .ASCIZ  /BAD= /
2133  012230     000
2134  012231     040  050040  036523  STMS:   .ASCIZ  /  PS= /
2135  012236  000040
2136  012240  005015  040504  040524  DAMS:   .ASCIZ  <15><12>/DATA= /
2137  012246  020075     000
2138  012251     015  005012  042524  ENDM1:  .ASCIZ  <15><12><12>/TEST COMPLETE/<15><12>
2139  012256  052123  041440  046517
2140  012264  046120  052105  006505
2141  012272  000012
2142  012274  050122  042101  036522  MRPADR: .ASCIZ  /RPADR=/
2143  012302     000
2144  012303     122  053120  041505  MRPVEC: .ASCIZ  /RPVEC=/
2145  012310  000075
2146  012312  050122  051120  047511  MRPPRI: .ASCIZ  /RPPRIO=/
2147  012320  000075
2148  012322  020040  020040  047117  ONLINE: .ASCIZ  /    ONLINE    /
```

```
2149  012330  044514  042516  020040
2150  012336  020040     000
2151  012341     040  020040  052440  UNSAFE: .ASCIZ  /    UNSAFE    /
2152  012346  051516  043101  020105
2153  012354  020040  000040
2154  012360  020040  020040  043117  OFFLIN: .ASCIZ  /    OFFLINE   /
2155  012366  046106  047111  020105
2156  012374  020040     000
2157  012377     122  030120  000062  RP02:   .ASCIZ  /RP02/
2158  012404  050122  031460     000  RP03:   .ASCIZ  /RP03/
2159
2160
2161          012412                          .EVEN
2162
2163                                  ;;*****************************************************************
2164
2165                                  ;BUFFERS START HERE
2166
2167          012412                  INBUF=.
2168          014712                  OUTBUF=INBUF+2300
2169
2170          000001                          .END
```

```
ADTYPE  010622          1877    1936#
ADT1    010746          1940    1947    1952    1955#
ATNB    001224           490#   1006    1717#   1722    1765
BAD     001202           481#   1902
BOMS    012223          1901    1986    2132#
BIT0  = 000001           392#
BIT00 = 000001           382#    392    1881    1936
BIT01 = 000002           381#    391    1884
BIT02 = 000004           380#    390    1872
BIT03 = 000010           379#    389    1869
BIT04 = 000020           378#    388    1866
BIT05 = 000040           377#    387    1875
BIT06 = 000100           376#    386    1878
BIT07 = 000200           375#    385     704     734     831     849     899     927
BIT08 = 000400           374#    384
BIT09 = 001000           373#    383    1683
BIT1  = 000002           391#
BIT10 = 002000           372#   1761
BIT11 = 004000           371#   1009    1689    1767
BIT12 = 010000           370#
BIT13 = 020000           369#   1694
BIT14 = 040000           368#   1687
BIT15 = 100000           367#    699     734     831     849     899     927    1691    1719
BIT2  = 000004           390#
BIT3  = 000010           389#
BIT4  = 000020           388#
BIT5  = 000040           387#
BIT6  = 000100           386#
BIT7  = 000200           385#
BIT8  = 000400           384#
BIT9  = 001000           383#
BPTVEC= 000014           399#
BUSADR  001176           479#    562*    1595    1597*
CHKAD   007760           743     856     938    1780#
CKSUM = 104406          1582#
CLRP    011216           680     720     738     778     835     853     874     903     931    1754    2019#
CLRS    004542           877     960#
CR    = 000015           307#   1409    1419
CRLF  = 000200           308#   1380    1419
CS    = 000020           411#    697     701     706     737     834     852     902     930    1011    1726    1763    1769
                        1773
CSMS    012127          1908    2118#
CSTYPE  010542          1868    1908#
CYL     001226           491#    726*    727     755*    756     799*    811     889*    890     911*    912     917*    918
                         948*    949    1794    1976
CYLA    001242           497#    785*    827     843
CYLR    001234           494#   1782*   1784*   1785*   1786*   1787*   1788*   1789*   1793*   1794    1988
CYMS    012162          1955    2124#
DA    = 000040           412#    697     737     834     852     902     930    1011
DAMS    012240          1915    2136#
DATA    001204           482#   1916
DATYPE  010556          1886    1915#
DD    = 000002           408#
DDISP = 177570           314#    460     582
DEBUF   001262           505#
DECCY   010734          1949    1953#
```

```
DECTK   010710          1945    1948#
DISPLA  001142           460#    582#   590#
DISPRE  000174           276#    590
DRVSTA  001274           519#    629    1673#  1674#  1675#  1676#  1685#  1693#
DRVTYP  001172           476#    635    1672#  1696#  1722
DS    = 000004           409#    697     701    706    737    834    852    902    930   1011   1726   1763   1769
                        1773
DSMS    012151          1929    2122#
DSTYPE  010606          1874    1929#
DSWR  = 177570           313#    459     581
DS1     007476          1714#   1716#
EMTVEC= 000030           402#    573#   574#
ENDM1   012251           759     858     951   2138#
ER    = 000010           410#    697     701    706    737    834    852    902    930   1011   1726   1763   1769
                        1773
ERMS    012140          1922    2120#
ERMS1   012052          1857    2108#
ERRVEC= 000004           395#    579     580#   591#  1651   1652#  1654#  1656#
ERTYPE  010572          1871    1922#
GB    = 000001           407#
GBTYPE  010514          1893    1898#
GBMS    012212          1868    1974    2130#
GETADR  006720           610    1595#
GMS   = ******  U        275     597     619    667   1575   1576   1577   1578   1580   1582   1583   1584   1585
                        1586    1587    1606   1620   1639   1660
GOOD    001200           480#   1899
GTSWR = 104405           606    1580#
HD    = 000100           413#   1800
HDMS    012010           780    1973    2100#
HDTYPE  011026          1830    1973#
HED     001230           492#    740#   750#   751    754#   803#   818#   819    820#   935#   944#   945    947#
                        1796    1979
HEDA    001244           498#    789#   828    844
HEDR    001236           495#   1781#  1782   1783#  1792#  1796   1991
HGL1    003230           822#    836
HGL2    003352           840#    854    857
HGO     003134           809#    851
HGO1    003336           833     837#
HGO2    003460           851     855#
HL1     002764           781#    784    788    791    793
HL2     003050           795#    798    802    805    807
HOME    007642           684     725    822    840    888    916   1754#  1764   1770   1774
HT    = 000011           305#   1378   1419
IL1     002252           681#
IL2     002334           691#    694
IL3     002346           693#    696
INBUF = 0124'2           730     742    809#   810#   811#   812#   813#   814#   815#   81'#   817#   819#   821#
                         824     837#   838#   839#   842    855    923    936   2167#  21'8
IOTVEC= 000020           400#
LERR    001260           504#
LF    = 000012           306#   1413   1419
MASK    001254           502#
MAXCYL  001170           475#    756    782    796    912    949   1721#  1724#
MKBUF   004332           887     985#
MRROR   012274          1598    2142#
MRTRI   012312          1626    2146#
```

```
MRPVEC  012303  1612  2144#
NEWMS   012113   795  2115#
NSTR    004504   879  1018#
NSWMES  011664   723  2085#
N52     002266   684#   698   702
N0      011341  1018  2038#
N1      011347  1019  2040#
N2      011355  1020  2042#
N3      011363  1021  2044#
N4      011371  1022  2046#
N5      011377  1023  2048#
N6      011405  1024  2050#
N7      011413  1025  2052#
N8      011421  1026  2054#
N9      011427  1027  2056#
OFFLIN  012360   641  2154#
OLDMS   012077   781  2112#
ONLINE  012322   632  2148#
OPTION  001174   478#   554#   556#   558#   560#   563#   652   655   658   66:   762
OUTBUF= 014712   895   906   986  2168#
PARIN   007570   783   787   797   801  1738#
PC     =000007  326#   600#   610#   611#   680#   681#   683#   684#   720#   721#   724#   725#   738#
                743#   778#   779#   783#   787#   790#   797#   801#   804#   822#   835#   840#   853#
                856#   874#   875#   877#   882#   883#   887#   888#   892#   903#   916#   920#   931#
                938#   965#   969#   970#   981#  1000#  1012#  1014#  1081#  1148#  1185#  1387#  1394#
               1401#  1415#  1417#  1655#  1701#  1706#  1715#  1725#  1734#  1738#  1740#  1748#  1750#
               1754#  1776#  1803#  1809#  1812#  1829#  1836#  1847#  1868#  1871#  1874#  1877#  1880#
               1893#  1886#  1904#  1911#  1918#  1925#  1932#  1969#  1997#  2007#  2015#  2021#
PIRQ   = 177772  312#
PIRQVE= 000240  406#
PR0    = 000000  329#
PR1    = 000040  330#
PR2    = 000100  331#
PR3    = 000140  332#
PR4    = 000200  333#
PR5    = 000240  334#
PR6    = 000300  335#
PR7    = 000340  336#   722   915
PS     = 177776  309#   310   679#   719#   777#   873#
PSW    = 177776  310#   690#   722#   915#
P.RVEC= 000024  401#
P2L1    000466  725#   765
P2L2    002476  727#   739   758
P2L3    002604  743#   748   753
P2N1    000570  736   740#
RDCHR  = 104407 1240  1583#  1834
RDLIN  = 104410 1314  1594#
RDOCT  = 104411 1585#  1608   1622   1641
RESREG= 104413 15 7#  1700   1887
RESVEC= 000010  396#
RIN     010242  683   724   969  1809  1834#
ROTOG   001256  503#  1841#  1843#
RPADR   001264  510#   651  1599  1611#  1653  1677
RPBA   = 000010  422#   730#   824#   842#   895#   923#
RPCA   = 000012  423#   727#   827#   843#   890#   918#  1938  1941
RPCS   = 000004  420#   687#   688#   689#   699   704   731#   732#   733#   734   825#   826#   830#
```

```
                        831    846*   847*   848*   849    896*   897*   898*   899    924*   925*   926*   927
                       1005*  1678*  1679*  1682*  1718*  1755*  1756*  1889   1909   2019*  2020*
RPDA   = 000014         424*   691    693    728*   828*   829*   844*   845*   891*   919*   1939   1942
RPDS   = 000000         418*  1004*  1006   1008*  1009   1683   1687   1689   1691   1694   1719   1761   1765
                       1767   1771   1775*  1930
RPER   = 000002         419*  1923   1936
RPI      002366         685    699*
RPINIT   007230         611   1668*
RPM1   = 000016         425*
RPM2   = 000020         426*
RPM3   = 000022         427*
RPPRIO   001272         512*  1627   1649*  1650   1671
RPVEC    001266         511*   685*  1613   1625*  1650*  1669
RPWC   = 000006         421*   686*   729*   823*   841*   894*   922*
RP02     012377         637   2157*
RP03     012404         639   2158*
RTL      010250        1715   1841*
R0     =%000000         317*   792    794    806    808    881*   971*   972    974    976    978*   979    1311
                       1315*  1316   1319   1339*  1342*  1370   1371*  1372   1375*  1516   1541*  1552    553*
                       1554   1555*  1556*  1557*  1558*  1609*  1611   1623*  1625   1627*  1628*  1629*  1630*
                       1631*  1632   1642*  1644*  1645*  1646*  1647*  1648*  1649   1708   1710   1711*  1717
                       1741   1810   1813   1815   1819*  1824   1828*  1835*  1845*  1863*  1864   2001   2003
                       2012
R1     =%000001         318*   621*   624    629    648*   1312   1317*  1325*  1327*  1329*  1332*  1335   1338*
                       1517   1540*  1651*  1654   1656   1669*  1670*  1671*  1680*  1682   1685*  1693*  1697*
R2     =%000002         319*   622*   635    649*   1313   1318*  1326*  1328*  1330*  1336   1337*  1518   1539*
                       1681*  1696   1698*
R3     =%000003         320*   691*   693    742*   855*   880*   936*   961*   962*   979*   987*   992    995*
                       1235   1237*  1238   1241*  1242   1249*  1250   1252   1260   1264   1266*  1272   1274
                       1276*  1279*  1453   1462*  1468*  1469*  1472*  1477*  1478*  1479   1488*  1519   1538*
                       1780*  1781   1790
R4     =%000004         321*   651*   686*   687*   688*   689*   691    693    699    704    727*   728*   729*
                        730*   731*   732*   733*   734    823*   824*   825*   826*   827*   828*   829*   830*
                        831    841*   842*   843*   844*   845*   846*   847*   848*   849    890*   891*   894*
                        895*   896*   897*   898*   899    918*   919*   922*   923*   924*   925*   926*   927
                       1004*  1005*  1006   1008*  1009   1454   1456*  1457*  1458*  1459   1460*  1474   1476*
                       1484*  1487*  1520   1537*  1677*  1678*  1679*  1682*  1683   1687   1689   1691   1694
                       1718*  1719   1755*  1756*  1761   1765   1767   1771   1775*  1889   1909   1923   1930
                       1936   1938   1939   1941   1942   2019*  2020*
R5     =%000005         322*   782*   786*   796*   800*   879*   881    906*   907*   908*   933*   937    942*
                        960*   963*   986*   990*   991*   992*  1455   1461*  1463*  1465*  1466*  1467*  1468
                       1486*  1521   1536*  1743   1858*  1859*  1860   1866   1869   1872   1875   1878   1881
                       1884
R6     =%000006         323*   325    567*   568*   569
R7     =%000007         324*   326
SAVREG= 104412         1586*  1668   1856
SCFLG    004554         976    978*  1041*
SCSLT    004530         890    933    942    961    987    995   1029*
SEC      001232         493*   741*   745*   746    749*   808*   810    937*  1798   1982
SECA     001246         499*   794*   829    845
SECMES   011251         876   2028*
SECR     001240         496*  1790*  1791*  1798   1994
SEKE     004430         892    920   1004*
SELTST   002064         615    651*
SEMS     012202        1964   2128*
SFCHK    003770         913    915*   921
```

```
SFCL1   004006          918#    932     950
SFCL2   004130          937#    941     946
SFCN1   004106          929     933#
SFSL1   003536          876#    884
SFSL2   003564          881#    886
SFSL3   003606          887#    893     954
SFSL4   003622          890#    904     914
SFSL5   003734          907#    910
SFSN1   003722          901     905#
SILO  = 000026          429#
SIN     004262          883     969#
SP    =X000006          325#    571#    579#    587#    591     624#    678#    702#    708#    718#    776#    872#    980#
                       1013#   1088#   1089#   1090    1099    1101    1104    1107#   1108#   1109    1114    1116    1118#
                       1119    1135#   1136#   1137    1147    1154#   1157#   1158#   1162#   1163#   1167    1170#   1174
                       1176    1178    1179#   1186    1188    1190#   1191    1193#   1194#   1195#   1196#   1197#   1212#
                       1213#   1214#   1215#   1216#   1222#   1235#   1236#   1241    1244    1248#   1255    1259#   1278
                       1279    1280#   1281#   1282#   1309#   1310#   1311#   1312#   1313#   1315    1319#   1321    1323
                       1331#   1332    1334    1335#   1337    1338    1339    1341    1370#   1371    1372#   1374    1375
                       1376#   1378    1380    1382    1388    1390#   1392#   1400#   1404    1408    1409    1413    1445#
                       1446    1447    1448#   1453#   1454#   1455#   1461    1486    1487    1488    1489#   1490#   1516#
                       1517#   1518#   1519#   1520#   1521#   1522#   1523#   1524#   1525#   1532#   1533#   1534#   1535#
                       1536    1537    1538    1539    1540    1541    1552#   1553    1563#   1564#   1599#   1609    1613#
                       1623    1632#   1642    1657    1749#   1757#   1758#   1760    1802#   1827#   1835    1841    1842#
                       1858    1860#   1863    1864#   1899#   1902#   1909#   1916#   1923#   1930#   1956#   1957#   1960#
                       1961#   1962#   1965#   1966#   1967#   1976#   1979#   1982#   1988#   1991#   1994#
SPACE   007554          1731#   1733    1748
STACK = 001100           300#    571     678     718     776     872
START   001356           555     557     559     561     564#
START1  001304           284     554#
START2  001314           287     556#
START3  001324           290     558#
START4  001334           293     560#
START5  001344           295     562#
STKLMT= 177774           311#
STMS    012231          1862    2134#
STR4    002450           708     722#
SUCA  = 000024           428#
SWMES   011435           682    2058#
SWR     001140           459#    569     581#    583     589#    604     614     760     859     952     1092    1131    1178#
                        1851    1854    1891
SWREG   000176           277#    589     604     1092    1131    1154
SW0   = 000001           364#
SW00  = 000001           354#    364
SW01  = 000002           353#    363
SW02  = 000004           352#    362
SW03  = 000010           351#    361
SW04  = 000020           350#    360
SW05  = 000040           349#    359
SW06  = 000100           348#    358
SW07  = 000200           347#    357     614
SW08  = 000400           346#    356
SW09  = 001000           345#    355
SW1   = 000002           363#
SW10  = 002000           344#   1851
SW11  = 004000           343#
SW12  = 010000           342#
```

```
SW13  = 020000            341#   1854
SW14  = 040000            340#    760     859     952
SW15  = 100000            339#   1891
SW2   = 000004            362#
SW3   = 000010            361#
SW4   = 000020            360#
SW5   = 000040            359#
SW6   = 000100            358#
SW7   = 000200            357#
SW8   = 000400            356#
SW9   = 001000            355#
SYSTAT  001704            613     616#
S1      012050            642    1731    1978    1981    1985    1990    1993    2107#
S2      012047           1975    1987    2106#
TAMS    012172           1959    2126#
TBITVE= 000014            397#
TEMP1   001252            501#
TEST1   002236            654     678#
TEST2   002430            657     718#
TEST3   002740            660     776#
TEST4   003516            663     872#
TKVEC = 000060            404#   1077*   1078*
TOG1    001206            483#    692*    695*    878*    885*    905*    909*    934*    940*    943*    985*    998*
TOG2    001210            484#    988*    993*    996*
TPVEC = 000064            405#
TRAPVE= 000034            403#    575*    576*
TRTVEC= 000014            398#
TTG     001250            500#   1732*   1745*   1746    1807*   1817*
TTIO    010122            790     804    1706    1738    1807#
TTO     011154            970    1812    2001#
TYPE  = 104401            595     599     617     643     645     647     664     665     669     682     723     759     780
                          781     795     858     876     951    1098    1152    1153    1156    1169    1180    1199    1247
                         1253    1258    1262    1267    1268    1270    1273    1277    1343    1345    1383    1480    1575#
                         1598    1604    1612    1618    1626    1637    1658    1705    1731    1826    1853    1857    1862
                         1888    1898    1901    1908    1915    1922    1929    1955    1959    1964    1973    1974    1975
                         1978    1981    1984    1985    1986    1987    1990    1993    1996    2004    2006    2013
TYPOC = 104402           1155    1576#   1861    1865    1900    1903    1910    1917    1924    1931    1958    1963    1968
                         1977    1980    1983    1989    1992    1995
TYPON = 104404           1578#
TYPOS = 104403            626    1577#   1601    1615    1634
TYPOUT  011204            882    2012#
UMES    011234           1705    2025#
UNIO    007426            681     721     779     875    1705#   1707    1709
UNIT    001222            489#    637     731     825     846     896     924    1710*   1712    1714    1718    1755
UNSAFE  012341            634    2151#
UNS1    007500           1713    1717#
URDY1   002256            682#    764
WORK    001212            485#    989*    991     997*
WORK1   001214            486#    785     789     799     803    1743    1808*   1821*   1822*   1823*   1824*   1828
WORK2   001216            487#   1939*   1942*   1943*   1944    1946*   1948    1950*   1951*   1953*   1961    1966
WORK3   001220            488#   1938*   1941*   1954*   1956
$AUTOB  001134            456#    608*   1145    1293
$.DAOR  001122            451#
$ DDAT  001126            453#
$.ELL   001160            469#   1098    1287    1853
$CHARC  006306           1385#   1395*   1402    1411*   1416#
```

```
$CKSWR  005040          1131#  1582
$CNTAG  001100           439#   566     567     577
$CNT3   = 000000         469#
$CNTLG  005702          1152   1288#
$CNTLU  005675          1169   1262    1287#
$CRLF   001165           471#   509     647     664     669    1180    1267    1287    1348    1384    1419    1826    1888
                        1984   1996    2006
$ERFLG  001103           442#
$ERMAX  001115           448#
$ERROR  010276           573    1851#
$ERRPC  001116           449#
$ERRTB  001304           542#
$ERTTL  001112           446#
$FILLC  001156           467#   1388    1419
$FILLS  001155           466#   1419
$GDADR  001120           450#
$GDDAT  001124           452#
$GTSWR  005130          1153#   1580
$HD     = 000000         258
$HIOCT  006070          1336*  1347#
$ICNT   001104           443#
$INTAG  001135           457#   1150*   1181    1293
$ITEMB  001114           447#
$LF     001166           472#   1277    1287    1348    1419
$LPADR  001106           444#
$LPERR  001111           445#
$MAIL   = ******  U       593    604     1372
$MNEW   005720          1156   1291#
$MSWR   005707          1153   1289#
$NULL   001154           465#   1390    1419
$OCNT   006534          1452*  1481*   1494#
$OMODE  006536          1447*  1451*   1456    1459*   1470*   1496#
$PASS   001100           440#
$QUES   001164           470#   1199    1270    1287    1345    1348    1419
$RDCHR  005342          1212#   1583
$RDDEC= ******  U       1586
$RDLIN  005432          1235#   1584
$RDOCT  005732          1309#   1585
$RDSZ   = 000007        1228#
$RESRE  006576          1531#   1587
$R2A    = ******  U     1588
$SAVRE  006540          1515#   1586
$SETUP= 000106           551#    572     573     575     577     601    1090    1095    1096    1126    1293
$STUP   = 177777         551#
$SWR    = 162000         247#    258     263     264     265     266     267     469     577
$TKB    001146           462#   1058    1079    1088    1107    1135    1162
$TKCNT  004566          1059#  1074*   1096    1113*   1219    1221*
$TKINT  004576           600    1074#   1148
$TKQEN= 004575          1063#   1121    1224
$TKQIN  004570          1060#  1075*   1076    1119*   1120*   1121    1123*
$TKQOU  004572          1061#  1076*   1222    1223*   1224    1226*
$TKQSR  004574          1062#  1075    1123    1226
$TKS    001144           461#   1058    1080*   1103*   1105    1111*   1133    1149*   1159    1183*
$TKSRV  004546          1077   1088#
$TM     = 000000         247#    258
$TPB    001152           464#   1408*   1419
```

```
$TPFLG  001157            468#   1366#  1419
$TPS    001150            463#   1406   1419
$TRAP   006634            575    1552#
$TRAP2  006656           1563#   1574
$TRP  = 000014           1567#   1576#  1577#  1578#  1579#  1580   1581#  1582   1583#  1584#  1585#  1586#  1587#
                         1598#
$TRPAD  006670           1557    1574#
$TSTNM  001102            441#
$TTYIN  005666           1237    1238   1250   1268   1282   1286#
$TYPBN= ****** U         1579
$TYPDS= ****** U         1579
$TYPE   006072           1366#   1367   1575
$TYPEC  006242           1185    1387   1394   1401   1406#  1407
$TYPEX  006310           1412    1414   1417#
$TYPOC  006336           1450#   1576
$TYPON  006352           1449    1452#  1578
$TYPOS  006312           1445#   1577
$DFILL  006535           1446*   1450*  1460   1495#
.       012412            271#    275#   281#   438#   473    570    598#   620#  1058   1062#  1063   1064#  1286#
                         1287    1293#  1348   1419   1661#  1739   1867   1870  1873   1876   1879   1882   1885
                         2161#   2167
```

```
COMMEN     1#    407#
ENDCOM     1#    407#
ERROR    301#    697     701     706     737     834     852     902     930    1011    1726    1763    1769    1773    1800
ESCAPE     1#    407#
GETPRI     1#    407#
GETSWR     1#    247#    407#    601
MORETA   431#    474
MULT       1#    407#
NEWTST     1#    407#
POP        1#    407#   1337    1536
PUSH       1#    407#   1311    1516
REPORT     1#    407#
SCOPE    302#
SETPRI     1#    407#   1215
SETTRA  1567#   1576    1577    1578    1580    1582    1583    1584    1585    1586    1587
SETUP      1#    407#    565
SKIP       1#    407#
SLASH      1#    407#
SPACE    407#
STARS      1#    407#    434     473     544     549    1057    1126    1141    1204    1228    1297    1351    1422    1500
        1546    2163
SWRSU      1#    407#    577#
TRMTRP  1567#
TYPBIN     1#    407#
TYPDEC     1#    407#
TYPNAM     1#    407#
TYPNUM     1#    407#
TYPOCS     1#    407#    623    1599    1613    1632
TYPOCT     1#    407#   1154    1976    1979    1982    1988    1991    1994
TYPTXT     1#    407#    595     616     665    1604    1618    1637    1658
SSCMRE   432#
SSCMTH   432#
SSESCA     1#    407#
SSNEXT     1#    407#
SSSET   1567#   1576    1577    1578    1580    1582    1583    1584    1585    1586    1587
SSSKIP     1#    407#
.EQUAT     1#    247#    297
.HEADE     1#    247#    248
.KT11      1#
.SETUP     1#    247#    551
.SWRHI     1#    247#    259
.SWRLO   267#
.$ACTI     1#
.$APTB     1#
.$APTH     1#
.$APTY     1#
.$ASTA     1#
.$CATC     1#    247#    269
.$CMTA     1#    247#    432
.$DB2D     1#
.$DB2O     1#
.$DIV      1#
.$EOP      1#
.$ERRO     1#
.$ERRT     1#
.$MULT     1#
```

```
.$POWE      1#
.$RAND      1#
.$RODE      1#
.$RODC      1#    247#   1295
.$READ      1#    247#   1055
.$R2AZ      1#
.$SAVE      1#    247#   1498
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    247#   1544
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#    247#   1349
.$TYPO      1#    247#   1420
.$40CA      1#
.1170       1#
```

```
            883   887   888   892   903   916   920   931   938   969   970  1148  1185  1387  1394
           1401  1706  1715  1738  1748  1754  1809  1812  1868  1871  1874  1877  1880  1883  1886
MOV         554   556   558   560   562   567   571   573   574   575   576   579   580   581   582
            587   589   590   591   622   624   632   634   637   639   641   642   651   678   685
            686   691   692   702   708   718   722   727   729   730   742   776   782   785   786
            789   794   796   799   800   803   808   810   811   823   824   627   841   842   843
            855   872   878   879   880   881   890   894   895   905   906   915   918   922   923
            933   934   936   937   942   943   960   961   979   985   986   987   988   991   992
            995   998  1075  1076  1077  1078  1080  1111  1123  1154  1178  1183  1212  1213  1216
           1226  1235  1237  1248  1279  1280  1281  1282  1309  1310  1311  1312  1313  1315  1316
           1335  1336  1337  1338  1339  1370  1371  1375  1390  1445  1453  1454  1455  1461  1468
           1486  1487  1488  1489  1490  1516  1517  1518  1519  1520  1521  1522  1523  1524  1525
           1532  1533  1534  1535  1536  1537  1538  1539  1540  1541  1552  1553  1557  1563  1564
           1599  1609  1611  1613  1623  1625  1627  1632  1642  1649  1650  1651  1652  1654  1656
           1669  1671  1677  1678  1679  1681  1710  1711  1714  1717  1721  1724  1757  1781  1792
           1790  1807  1828  1835  1841  1858  1860  1863  1864  1899  1902  1909  1916  1923  1930
           1938  1939  1941  1942  1953  1956  1976  1979  1982  1988  1991  1994  2012  2019  2020
MOVB        608   687   689   731   733   825   828   829   830   844   845   846   848   896   898
            924   926  1004  1005  1013  1013  1107  1119  1135  1150  1162  1222  1241  1246  1252
           1257  1272  1319  1372  1400  1408  1446  1447  1450  1451  1452  1456  1459  1460  1479
           1555  1682  1685  1693  1718  1755  1756  1775  1961  1966  2003
NEG        1457
NOP         744   939
RESET       564
ROL         812   813   814   815   816   817   818  1326  1328  1330  1463  1465  1466  1467  1469
           1821  1822  1823
ROR         820  1783  1784  1785  1786  1787  1788  1789
RTI         5 3   709  1112  1124  1184  1217  1227  1283  1340  1377  1491  1526  1542  1565  1894
RTS         965   981  1000  1012  1014  1081  1417  1558  1655  1701  1725  1734  1740  1750  1776
           1803  1829  1836  1847  1904  1911  1918  1925  1932  1969  1997  2007  2015  2021
SUB         971  1859
SWAB       1631
TRAP       1567  1576  1577  1578  1580  1582  1583  1584  1585  1586  1587
TST         602   974  1079  1099  1104  1147  1176  1191  1219  1244  1255  1278  1334  1341  1374
           1382  1404  1474  1554  1595  1653  1712  1760
TSTB        629   976  1105  1133  1159  1346  1406  1889
.ASCII      470   471  2058  2064  2070  2076  2106
.ASCIZ      469   472   598   620   668  1287  1298  1289  1291  1607  1621  1640  1661  2025  2028
           2038  2040  2042  2044  2046  2048  2050  2052  2054  2056  2078  2093  2100  2107  2108
           2112  2115  2118  2120  2122  2124  2126  2128  2130  2132  2134  2136  2138  2142  2144
           2146  2148  2151  2154  2157  2158
.BLKB      1062  1286
.BYTE       441   442   447   448   456   457   465   466   467   468   519   520   521   522   523
            524   525   526   627   628  1042  1043  1044  1045  1046  1047  1048  1049  1050  1051
           1284  1285  1492  1493  1494  1495  1602  1603  1616  1617  1635  1636  2008
.DSABL     1281
.EN BL        1   247  1058
.END       2170
.ENDC       253   266   267   278   301   393   407   435   439   441   469   470   474   528   545
            550   551   571   572   573   575   577   593   598   604   610   620   628   629   668
           1058  1087  1090  1094  1125  1127  1142  1165  1201  1205  1216  1228  1229  1237  1239
           1242  1270  1287  1293  1298  1304  1348  1352  1372  1423  1501  1547  1553  1556  1575
           1576  1577  1578  1579  1580  1581  1582  1583  1584  1585  1586  1587  1588  1603  1604
           1607  1617  1618  1621  1636  1637  1640  1661  2164
.EQUIV      301   302   310   325   326   355   356   357   358   359   360   361   362   363   364
            383   384   385   386   387   388   389   390   391   392
```

```
.EVEN    598   620   668  1064  1293  1607  1621  1640  1661  2161
.IF      249   266   267   278   299   365   393   434   438   440   469   473   474   544   549
         551   566   571   573   575   577   593   597   601   604   619   627   628   667  1057
        1059  1087  1090  1126  1127  1141  1165  1204  1205  1215  1228  1236  1238  1242  1243
        1296  1297  1293  1297  1300  1316  1351  1372  1422  1500  1546  1552  1556  1567  1576
        1577  1578  1579  1580  1582  1583  1584  1585  1586  1587  1588  1602  1603  1606  1616
        1617  1620  1635  1636  1639  1660  2163
.IFF     266   267   299   435   438   440   469   474   545   550   571   628  1058  1127  1142
        1152  1205  1208  1216  1228  1229  1238  1270  1286  1298  1352  1423  1501  1547  1553
        1603  1604  1617  1618  1636  1637  2164
.IFT     598   620   668  1207  1212  1321  1341  1348  1607  1621  1640  1661
.IFTF    598   620   668  1152  1205  1208  1317  1325  1347  1607  1621  1640  1661
.IIF     248   253   258   263   264   265   266   267   268   275   473   572   577   625  1058
        1064  1095  1096  1155  1278  1287  1293  1348  1419  1575  1576  1577  1578  1580  1582
        1583  1584  1585  1586  1587  1600  1614  1633  1977  1980  1983  1989  1992  1995
.IRP     474   551  1311  1337  1516  1536
.LIST      1   247   275   407   469   551   577   598   620   668  1228  1567  1575  1576  1577
        1578  1579  1580  1581  1582  1583  1584  1585  1586  1587  1588  1607  1621  1640  1661
.MACRO     1   267   431   432  1567
.MCALL   247   407   577
.NLIST     1   247   275   407   469   551   577   598   620   668  1228  1567  1575  1576  1577
        1578  1579  1580  1581  1582  1583  1584  1585  1586  1587  1588  1607  1621  1640  1661
.PAGE    432   528
.PSECT     1
.RADIX   275  1029  1041  1628  1644
.SBTTL   259   269   279   297   432   528   552   565   601   673   711   770   866  1053  1055
        1295  1349  1420  1498  1544  1567  2023
.TITLE   248
.WORD    275   276   277   440   443   444   445   446   449   450   451   452   453   454   455
         458   459   460   475   476   478   479   480   481   482   483   484   485   486   487
         488   489   490   491   492   493   494   495   496   497   498   499   500   501   502
         503   504   505   510   511   512   644   646  1018  1019  1020  1021  1022  1023  1024
        1025  1026  1027  1030  1031  1032  1033  1034  1035  1036  1037  1038  1039  1059  1060
        1061  1344  1347  1416  1496  1574  1716  2014


ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,MOM.SEQ/SOL/CRF/ML:TOC/PAGNUM=DZRP2B.SML,DZRP2B.CMB
RUN-TIME: 28 35 4 SECONDS
RUN-TIME RATIO: 154/69=2.2
CORE USED:  33K  (65 PAGES)
```

Spooler runtime 0 Seconds, 39 KCS, 830 disk reads, 3 disk writes, 86 pages