TR79F

TRDP/XXDP TR79F MONITOR

MD-11-DMQUF-A

EP-DMQUF-A-DL-A

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DMQUF-A-LA

PRODUCT NAME: TRDP - XXDP TR79F MONITOR

DATE: JANAURY 1977

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR(S): DIAGNOSTIC ENGINEERING

%

```
                                .LIST   BIN
                                .SBTTL  EQUATES
000000                          R0      =%0
000001                          R1      =%1
000002                          R2      =%2
000003                          R3      =%3
000004                          R4      =%4
000005                          R5      =%5
000006                          R6      =%6
000006                          SP      =%6
000007                          PC      =%7
000000                          OPEN    =0
177776                          PS      =177776
177776                          PSW     =177776
000340                          PRTY7   =340
177716                          XWCTR   =-62    ;INDEX TO WRITE COUNTER.
177720                          XFLMOD  =-60    ;INDEX TO FILE MODE INDICATOR
177722                          XFLCNT  =-56    ;INDEX TO FILE COUNT
177724                          XSVMAP  =-54    ;
177726                          XSVCNT  =-52    ;
177730                          XSVBLK  =-50    ;
177732                          XSVNAM  =-46    ;PHONY UFD BLOCK POINTERS
177736                          XSVEXT  =-42    ;
177740                          XSVDAT  =-40    ;
177742                          XSVXX   =-36    ;
177744                          X1STBK  =-34    ;
177746                          XBKLGT  =-32    ;
177750                          XLSTBK  =-30    ;
177752                          XSVUPT  =-26    ;
177754                          XBT     =-24    ;INDEX TO BOOT ROUTINE.
177756                          DRT     =-22    ;INDEX TO DIRECTORY ROUTINE
177760                          ZER     =-20    ;INDEX TO ZERO ROUTINE
177762                          DLT     =-16    ;INDEX TO DELETE ROUTINE
177764                          CLS     =-14    ;INDEX TO CLOSE ROUTINE
177766                          ENTR    =-12    ;INDEX TO ENTER ROUTINE
177770                          SRH     =-10    ;INDEX TO LOKUP ROUTINE
177772                          ALC     =-6     ;INDEX TO ALLOCATE ROUTINE
177774                          XSV     =-4     ;INDEX TO SERVICE ROUTINE (DRIVER).
177776                          XDN     =-2     ;DRIVE NUMBER INDEX
000000                          XCM     =0      ;INDEX TO COMMAND REGISTER
000002                          XWC     =2      ;INDEX TO WORD COUNT
000004                          XBA     =4      ;INDEX TO BUS ADDRESS
000006                          XDT     =6      ;INDEX TO BLOCK NUMBER
000010                          XCO     =10     ;INDEX TO COMMAND
000012                          XRD     =12     ;INDEX TO READ COMMAND
000014                          XWT     =14     ;INDEX TO WRITE COMMAND
000016                          XBC     =16     ;INDEX TO REQUESTED BLOCK COUNT
000020                          XDR     =20     ;INDEX TO 1ST DIR BLOCK POINTER.
000022                          XNB     =22     ;INDEX TO LAST BLOCK # ALLOCATED.
000024                          XXNAM   =24     ;INDEX TO ASCII NAME IN DDB

010000                          MONCNT  =10000          ;MONITOR SIZE IS 4K
000024                          MNBK    =24             ;POINTER TO MONITOR CORE IMAGE.

000033                          ALTMOD  =33
000033                          ALT1    =33
```

```
                000175                  ALT2    =175
                000176                  ALT3    =176


                                        .SBTTL  GLOBAL REFERENCE DEFINITIONS

                                ;EXTERNAL GLOBAL DEFINITIONS

                                        .GLOBL  SETMTO
                                        .GLOBL  CREWND,CRDHDR


                000000                  .ASECT
                164002                  MTST    =164002
                164000                  MTCM    =164000
                164004                  MTWC    =164004
                164006                  MTBA    =164006
```

```
                                        .SBTTL  ONCE ONLY BOOT LOADER

                000000                  .=0
000000  000240                          240
000002  012700  036000                  MOV     #36000,R0               ;RELOCATION ADDRESS
000006  005001                          CLR     R1                      ;INITIALIZE TO OLD ADDRESS
000010  012702  000400                  MOV     #256.,R2                ;SET COUNT OF TRANSFERS
000014  112120                  1$:     MOVB    (R1)+,(R0)+             ;MOVE BYTE TO LPER CORE
000016  005302                          DEC     R2                      ;COUNT
000020  001375                          BNE     1$                      ;LOOP BACK IF NOT DONE
000022  000137  036070                  JMP     @#36070                 ;JUMP TO BOOT LOAD SUB ROUTIN
                000040                  .=40
000040  000000  000000  000000          .WORD   0,0,0,0,0,0,0,0,0,0,0,0
000046  000000  000000  000000
000054  000000  000000  000000
000062  000000  000000  000000



                000070                  .=70
000070  000240                  INIT:   NOP                             ;RESET PERIPHERALS
000072  012706  040000                  MOV     #40000,SP               ;INITIALIZE STACK POINTER
                                ;CHECK IF MTU READY AND AT LOAD POINT
000076  004767  000160                  JSR     PC,READY                ;TEST FOR DONE
000102  032737  000040  164002          BIT     #40,@#MTST                      ;TEST FOR LOAD POINT INDICATOR
                                ;IF YES READ ID BURST, IF NO REWIND TAPE TO LOAD POINT
000110  001007                          BNE     BURST                   ;BRANCH IF @ LOAD POINT
000112  005037  164002          RWND:   CLR     @#MTST                  ;CLEAR STATUS REG.
000116  012737  000021  164000          MOV     #21,@#MTCM                      ;REWIND COMMAND
000124  004767  000132                  JSR     PC,READY                ;WAIT FOR DONE
000130  005037  164002          BURST:  CLR     @#MTST                  ;CLEAR STATUS REG.
000134  012737  000005  164000          MOV     #5,@#MTCM                       ;READ COMMAND
000142  004767  000114                  JSR     PC,READY                ;DONE??
000146  032737  000020  164002          BIT     #20,@#MTST                      ;IDB BIT INDICATOR SET ??
                                ;IO BURST IS TESTED FOR, IF NOT PRESENT AT BEGINNING OF TAPE
                                ;THE TAPE HALTS.
000154  001454                          BEQ     TAPERR
000156  005037  164002                  CLR     @#MTST                  ;CLEAR STATUS REG.
000162  012737  000005  164000          MOV     #5,@#MTCM               ;SET READ FUNCTION
000170  004767  000066                  JSR     PC,READY                ;WAIT FOR DONE
000174  005004                          CLR     R4                      ;SET INITIAL LOAD ADDRESS TO 0
000176  010437  164006          REED:   MOV     R4,@#MTBA                       ;R4 HOLDS INITIAL LOOP ADDRESS DURING DMA
000202  012737  177000  164004          MOV     #-512.,@#MTWC           ;SET BLOCK SIZE TO READ 256. (DECIMAL)
                                                                        ;CHARACTERS
000210  005037  164002                  CLR     @#MTST
000214  012737  000005  164000          MOV     #5,@#MTCM
000222  004767  000034                  JSR     PC,READY
000226  032737  000010  164002          BIT     #10,@#MTST                      ;CHECK IF END OF FILE MARK HAS BEEN READ
000234  001402                          BEQ     3$
000236  000137  001000                  JMP     @#1000                  ;JUMP TO THE BEGIN OF TRDP PROG.

                                ;CHARACTERS HAVE BEEN TRANSFERRED FROM TAPE TO CORE, BEGIN TO
                                ;PACK THE CHARACTERS INTO WORDS.

000242  012703  177000          3$:     MOV     #-512.,R3               ;COUNTER
```

```
(2)  000246  010405                    MOV     R4,R5           ;R4 USED TO POINT TO WORD ADDRESSES
(2)  000250  112524                    MOVB    (R5)+,(R4)+     ;R5 USED TO POINT TO BYTE ADDRESSES
(2)  000252  005205                    INC     R5
(2)  000254  005203                    INC     R3              ;COUNT # OF WORDS
(2)  000256  001374                    BNE     PACK            ;IF NOT, KEEP PACKING
(2)  000260  000746                    BR      REED            ;IF YES, GO BACK AND GET ANOTHER
(2)                                                            ;BLOCK FROM TAPE
(2)
(2)  000262  105737  164000            TSTB    @#MTCM          ;CHECK IF TAPE CONTROLLER READY
(2)  000266  100375                    BPL     READY           ;WAIT UNTIL READY
(2)  000270  005737  164000            TST     @#MTCM          ;CHECK IF ERROR BIT SET
(2)  000274  100005                    BPL     RTN             ;IF NO THEN RETURN
(2)  000276  032737  011000  164002    BIT     #11000,@#MTST   ;IF YES, THEN CHECK IF EITHER A
(2)                                                            ;READ COUNT OF SINGLE TRACK ERROR
(2)                                                            ;HAS OCCURRED.
(2)  000304  001001                    BNE     RTN
(2)  000306  000000                    HALT                    ;HALT BECAUSE OF TAPE ERROR
(2)  000310  000207                    RTS     PC              ;IF YES, IGNORE AND RETURN
```

```
(1)                                             .SBTTL  SIZER, MONITOR RELOCATOR, AND OTHER GOODIES.
(1)             001000                          .=1000
(1)  001000  012706  000050'           BEGIN:   MOV     #SPBOT,R6          ;SET UP STACK.
(1)                                     ;SIZE CORE AND UPDATE THE LITERALS OF RESIDENT MONITOR.
(1)  001004  012767  001072  176772             MOV     #3$,4              ;POINT TIMEOUT TRAP TO 3$
(1)  001012  012700  174000                     MOV     #-4000,R0          ;DETERMINE TOP OF CORE IN 1K CHUNKS.
(1)  001016  005001                             CLR     R1
(1)  001020  062700  004000            1$:      ADD     #4000,R0
(1)  001024  062710  000000                     ADD     #0,(0)             ;REFERENCE UNKNOWN LOC.
(1)  001030  005201                             INC     R1                 ;IF HERE, NO TRAP OCCURRED.
(1)  001032  022701  000034                     CMP     #28.,R1            ;DONE 28 TIMES?
(1)  001036  001370                             BNE     1$                 ;BR IF NOT.
(1)  001040  062700  004000                     ADD     #4000,R0
(1)  001044  012767  000006  176732    2$:      MOV     #6,4               ;RESTORE ERROR TRAP.
(1)  001052  020127  000010                     CMP     R1,#8.             ;8K OR GREATER?
(1)  001056  002010                             BGE     4$                 ;BR IF YES.
(1)  001060  004567  000642'                    JSR     R5,MES             ;INSUFFICIENT CORE MESSAGE.
(1)  001064  001435                             NOCORE
(1)  001066  000000                             HALT
(1)  001070  000777                             BR      .                  ;LOCK IN HALT.
(1)  001072  012716  001044            3$:      MOV     #2$,(6)            ;TRAPPED TO HERE. EXIT TO 2$
(1)  001076  000002                             RTI
(1)  001100  006301                    4$:      ASL     R1                 ;READY TO TYPE CORE SIZE.
(1)  001102  116167  001236  000302             MOVB    KCODE(1),AK+1
(1)  001110  116167  001237  000275             MOVB    KCODE+1(1),AK+2
(1)  001116  166700  000112                     SUB     LIMIT+2,R0         ;SET UP NEW LOAD ADDRESS
(1)  001122  010001                             MOV     R0,R1              ;
(1)  001124  010167  000652'                    MOV     R1,RELCNT          ;SAVE IT AT RELTMP
(1)  001130  012702  002624                     MOV     #$LITTB,R2         ;
(1)  001134  022712  177777            51$:     CMP     #-1,(R2)           ;END OF TABLE?
(1)  001140  001402                             BEQ     6$                 ;BR IF YES.
(1)  001142  060132                             ADD     R1,@(R2)+          ;CORRECT FOR RELOCATION.
(1)  001144  000773                             BR      51$
(1)                                     ;RELOCATE MONITOR TO TOP OF CORE.
(1)  001146  005001                    6$:      CLR     R1                 ;WHERE PROGRAM STARTS.
(1)  001150  016702  000060                     MOV     LIMIT+2,R2
(1)  001154  112120                    7$:      MOVB    (1)+,(0)+          ;RELOCATE ONE BYTE AT A TIME.
(1)  001156  005302                             DEC     R2                 ;DONE?
(1)  001160  001375                             BNE     7$                 ;CONTINUE RELOCATION.
(1)  001162  004567  000642'                    JSR     R5,MES             ;TYPE TITLE AND CORE SIZE.
(1)  001166  001330                             NAME
(1)  001170  004567  000642'                    JSR     R5,MES             ;TYPE RESTART ADDR.
(1)  001174  001416                             ARSTRT
(1)  001176  016703  000136'                    MOV     $COMC3,R3
(1)  001202  004767  001232'                    JSR     PC,ITOA
(1)  001206  004567  000642'                    JSR     R5,MES             ;TYPE FOR HELP MESSAGE.
(1)  001212  001460                             FHELP
(1)  001214  004567  000642'                    JSR     R5,MES             ;TYPE THE HELP MESSAGE
(1)  001220  001547                             MNINST
(1)  001222  004767  000202'                    JSR     PC,DELAY           ;WAIT A BIT.
(1)  001226  016707  000136'                    MOV     $COMC3,PC          ;GOTO RESIDENT MONITOR.
(1)  001232  000000  000000            LIMIT:   .LIMIT
(1)                                             .EVEN
(1)  001236  030040  030440  031040    KCODE:   .ASCII  ' 0 1 2 3 4 5 6 7 8 9'
(1)  001244  031440  032040  032440
(1)  001252  033040  033440  034040
```

```
(1)  001260  034440
(1)  001262  030061  030461  031061          .ASCII  '10111213141516171819 20'
(1)  001270  031461  032061  032461
(1)  001276  033061  033461  034061
(1)  001304  034461  030062
(1)  001310  030462  031062  031462          .ASCII  '21222324252627 28'
(1)  001316  032062  032462  033062
(1)  001324  033462  034062
(1)  001330  052045  042122  020120  NAME:   .ASCII  "%TRDP - XXDP TR79F MONITOR M-11-DMQUF-A 21-OCT-76"
(1)  001336  020055  054130  050104
(1)  001344  052040  033522  043071
(1)  001352  046440  047117  052111
(1)  001360  051117  046440  030455
(1)  001366  026461  046504  052521
(1)  001374  026506  020101  030462
(1)  001402  047455  052103  033455
(1)  001410     066
(1)  001411     040  020040  000113  AK:     .ASCIZ  '  K'
(1)  001416  051045  051505  040524  ARSTRT: .ASCIZ  '%RESTART ADDR:'
(1)  001424  052122  040440  042104
(1)  001432  035122     000
(1)  001435     045  047111  052523  NOCORE: .ASCIZ  '%INSUFFICIENT CORE'
(1)  001442  043106  041511  042511
(1)  001450  052116  041440  051117
(1)  001456  000105
(1)  001460  052045  020117  041101  FHELP:  .ASCIZ  '%TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C (↑C)%'
(1)  001466  051117  020124  044124
(1)  001474  020105  047506  046114
(1)  001502  053517  047111  020107
(1)  001510  042510  050114  046440
(1)  001516  051505  040523  042507
(1)  001524  052040  050131  020105
(1)  001532  052103  046122  041440
(1)  001540  024040  041536  022451
(1)  001546     000
(1)  001547     045  052045  050131  MNINST: .ASCII'%%TYPE:'
(1)  001554  035105
(1)  001556  043045  041474  037122          .ASCII  '%F<CR> TO SET CONSOLE FILL COUNT'
(1)  001564  052040  020117  042523
(1)  001572  020124  047503  051516
(1)  001600  046117  020105  044506
(1)  001606  046114  041440  052517
(1)  001614  052116
(1)  001616  042045  041474  037122          .ASCII  '%D<CR> FOR DIRECTORY ON CONSOLE, OR'
(1)  001624  043040  051117  042040
(1)  001632  051111  041505  047524
(1)  001640  054522  047440  020116
(1)  001646  047503  051516  046117
(1)  001654  026105  047440     122
(1)  001661     045  027504  036106          .ASCII  '%D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR'
(1)  001666  051103  020076  047506
(1)  001674  020122  044123  051117
(1)  001702  020124  044504  042522
(1)  001710  052103  051117  020131
(1)  001716  047117  041440  047117
(1)  001724  047523  042514  020054
```

```
(1)  001732  051117
(1)  001734  042045  046057  041474          .ASCII  '%D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR'
(1)  001742  037122  043040  051117
(1)  001750  042040  051111  041505
(1)  001756  047524  054522  047440
(1)  001764  020116  044514  042516
(1)  001772  050040  044522  052116
(1)* 002000  051105  020054  051117
(1)  002006  042045  046057  043057          .ASCII  '%D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,'
(1)  002014  041474  037122  043040
(1)  002022  051117  051440  047510
(1)  002030  052122  042040  051111
(1)  002036  041505  047524  054522
(1)  002044  047440  020116  044514
(1)  002052  042516  050040  044522
(1)  002060  052116  051105     054
(1)  002065     045  020122  047503          .ASCII  '%R COPY<CR> TO RUN COPY PROGRAM,'
(1)  002072  054520  041474  037122
(1)  002100  052040  020117  052522
(1)  002106  020116  047503  054520
(1)  002114  050040  047522  051107
(1)  002122  046501     054
(1)  002125     045  020122  044506          .ASCII  '%R FILENAME<CR> TO RUN ANY OTHER PROGRAM.'
(1)  002132  042514  040516  042515
(1)  002140  041474  037122  052040
(1)  002146  020117  052522  020116
(1)  002154  047101  020131  052117
(1)  002162  042510  020122  051120
(1)  002170  043517  040522  027115
(1)  002176  046045  043040  046111          .ASCII  '%L FILENAME<CR> TO LOAD A PROGRAM ONLY'
(1)  002204  047105  046501  036105
(1)  002212  051103  020076  047524
(1)  002220  046040  040517  020104
(1)  002226  020101  051120  043517
(1)  002234  040522  020115  047117
(1)  002242  054514
(1)  002244  051445  041474  037122          .ASCII  '%S<CR> TO START THE PROGRAM JUST LOADED,'
(1)  002252  052040  020117  052123
(1)  002260  051101  020124  044124
(1)  002266  020105  051120  043517
(1)  002274  040522  020115  052512
(1)  002302  052123  046040  040517
(1)  002310  042504  026104
(1)  002314  051445  040440  042104          .ASCII  '%S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS'
(1)  002322  036122  051103  020076
(1)  002330  047524  051440  040524
(1)  002336  052122  052040  042510
(1)  002344  050040  047522  051107
(1)  002352  046501  040440  020124
(1)  002360  050123  041505  043111
(1)  002366  041511  040440  042104
(1)  002374  042522  051523
(1)  002400  041445  043040  046111          .ASCII  '%C FILENAME<CR> TO RUN A CHAIN,'
(1)  002406  047105  046501  036105
(1)  002414  051103  020076  047524
(1)  002422  051040  047125  040440
```

```
(1)  002430  041440  040510  047111
(1)  002436     054
(1)  002437     045  020103  044506          .ASCII  '%C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.'
(1)  002444  042514  040516  042515
(1)  002452  050457  036126  051103
(1)  002460  020076  047524  051040
(1)  002466  047125  040440  041440
(1)  002474  040510  047111  044440
(1)  002502  020116  052521  041511
(1)  002510  020113  042526  044522
(1)  002516  054506  046440  042117
(1)  002524  027105
(1)  002526  051045  043105  051105          .ASCIZ  '%REFER TO XXDP USER MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.%'
(1)  002534  052040  020117  054130
(1)  002542  050104  052440  042523
(1)  002550  020122  040515  052516
(1)  002556  046101  046440  026504
(1)  002564  030461  042055  050532
(1)  002572  040530  043040  051117
(1)  002600  040440  042104  052111
(1)  002606  047511  040516  020114
(1)  002614  042510  050114  022456
(1)  002622     000
(1)          002624                          .EVEN
(1)

(1)  002624  000154'                  $LITTB: $REL1
(1)  002626  001570'                          $REL2+2
(1)  002630  001550'                          $REL3
(1)  002632  000332'                          $REL4
(1)  002634  000334'                          $REL4+2
(1)  002636  000270'                          $REL5
(1)  002640  000272'                          $REL5+2
(1)  002642  000250'                          $REL6
(1)  002644  000252'                          $REL6+2
(1)  002646  001510'                          $REL7
(1)  002650  002540'                          $REL10+2
(1)  002652  000054'                          $REL11+2
(1)  002654  000374'                          $REL12+2
(1)  002656  000342'                          $REL13+2
(1)  002660  002276'                          $REL14+2
(1)  002662  002152'                          $REL15+2
(1)  002664  001462'                          $REL16+2
(1)  002666  000136'                          $COMC3
(1)  002670  000146'                          COMCON+2
(1)  002672  001032'                          GETINO+2
(1)  002674  003120'                          LOAD4+2
(1)  002676  002102'                          $BUF
(1)  002700  002072'                          $BUF2
(1)  002702  002156'                          $TXNAM
(1)  002704  001452'                          $IDDB
(1)  002706  177777                           -1
(1)
(1)          000000'                          .OSECT  MTDIRT
```

```
                                        .SBTTL  NON-RESIDENT CODE
          000000'                       .CSECT
                                ;DIRECTORY ROUTINE.
000000' 005726                  NRDIR:  TST     (SP)+           ;POP OFF 1ST STACK ELEMENT.
000002' 012667  002410'                 MOV     (SP)+,FILLCT    ;GET FILL COUNT.
000006' 012667  001466'                 MOV     (SP)+,CURDRV    ;GET THE CURRENT DRIVE NUMBER.
000012' 012667  000136'                 MOV     (SP)+,$COMC3    ;GET RES MONITOR RESTART ADDR.
000016' 012667  001124'                 MOV     (SP)+,KBPTR     ;GET KYBD POINTER.
000022' 005002                          CLR     R2              ;FOR DEVICE SET ROUTINE.
000024' 004767  001442'                 JSR     PC,SETI         ;SET INPUT. NO NAME.
000030' 105067  000030                  CLRB    FSTMOD          ;ASSUME NO FAST MODE SWITCH.
000034' 004767  000034                  JSR     PC,NRGTSW       ;GET SWITCHES.
000040' 004567  002114'                 JSR     R5,BCLEAR       ;CLEAR NAME TO 0'S
000044' 004506' 000077  000011          IFNAM,77,9.
000052' 010565  177720                  MOV     R5,XFLMOD(R5)   ;INDICATE WILD MODE.
000056' 004775  177756                  JSR     PC,@DRT(R5)     ;GO OUTPUT DIRECTORY.
000062' 000207                          RTS     PC              ;RETURN.

000064'    000                  FSTMOD::        .BYTE   0
          000066'                       .EVEN

000066' 105267  177772          SETFST: INCB    FSTMOD          ;SET FAST MODE.
000072' 000400                          BR      NRGTSW

000074' 012701  000144'         NRGTSW: MOV     #NRSWTB,R1      ;POINT TO NON-RESIDENT SWITCH TABLE.
000100' 000167  000552'                 JMP     GTOKK           ;GO SET SWITCHES.

000104' 016767  000030  001004' LPSW:   MOV     LPS,MREG        ;CHANGE STATUS REG.
000112' 016767  000024  001000'         MOV     LPB,MOUT        ;CHANGE BUFFER REG.
000120' 052767  000200  002410'         BIS     #200,FILLCT     ;MAKE FILLCOUNT NEGATIVE.
000126' 012702  000014                  MOV     #14,R2          ;OUTPUT A FORM FEED.
000132' 004767  000202'                 JSR     PC,DELAY        ;WAIT FOR FORM FEED DONE.
000136' 000756                          BR      NRGTSW          ;SEE ABOUT OTHER SWITCHES.
000140' 177514                  LPS:    177514                  ;LINE PRINTER STAT REG ADDR.
000142' 177516                  LPB:    177516                  ;LINE PRINTER BUFFER REG ADDR.

000144'                         NRSWTB:
000144' 000066'                         .WORD   SETFST          ;DISPATCH ADDRESS FOR 'F
000154' 000104'                         .WORD   LPSW            ;DISPATCH ADDRESS FOR 'L
000164' 000212'                         .WORD   RTSPC           ;DISPATCH ADDRESS FOR '15
000172' 000546'                         .WORD   GTOK            ;DISPATCH ADDRESS FOR '40
000200' 177777                          -1                      ;TERMINATOR.
```

```
(1)                                     ;SUB TO TYPE FILE NAME.
(2)   000202'                           $TPNM1::
(2)   000202'  004467  002040'                  JSR     R4,SAV04        ;SAVE REGS 0-4
(1)   000206'  016703  002156'                  MOV     $TXNAM,R3       ;ADDR OF NAME TO R3.
(1)   000212'  005001                           CLR     R1              ;CHAR COUNTER.
(1)   000214'  112302                  1$:     MOVB    (R3)+,R2        ;GET A CHAR.
(1)   000216'  004767  000666'         2$:     JSR     PC,MES1         ;TYPE IT.
(1)   000222'  005201                           INC     R1              ;UP CHAR COUNT.
(1)   000224'  020127  000006                   CMP     R1,#6           ;DONE 6?
(1)   000230'  103771                           BLO     1$              ;BR IF NOT YET.
(1)   000232'  101003                           BHI     3$              ;BR IF MORE THAN 6.
(1)   000234'  112702  000056                   MOVB    #'.,R2          ;SIX. TYPE A DOT.
(1)   000240'  000766                           BR      2$              ;GO TYPE THE DOT.
(1)   000242'  120127  000012          3$:     CMPB    R1,#10.         ;DONE 10?
(1)   000246'  103762                           BLO     1$              ;BR IF NOT.
(2)   000250'  004767  002052'                  JSR     PC,RST04        ;RESTORE REGS 0-4
(1)   000254'  000167  000752'                  JMP     TAB             ;GO TAB AND RTS PC.
```

```
(1)                                             .SBTTL  BINARY TO DECIMAL CONVERT AND TYPE SUBROUTINE
(2)  000260'                            BCDCV::
(2)  000260'  004467  002040'                   JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  000264'  012504                            MOV     (R5)+,R4                ;NUMBER OF DIGITS
(1)  000266'  012700  000374'                   MOV     #DECTAB,R0              ;TABLE OF DECIMAL NUMBERS
(1)  000272'  005046                            CLR     -(SP)                   ;STACK WORD FOR INDICATOR.
(1)  000274'  005740                    BCD1:   TST     -(R0)                   ;STEP TO THE LARGEST DIGIT
(1)  000276'  005304                            DEC     R4
(1)  000300'  003375                            BGT     BCD1
(1)  000302'  005002                    BCD2:   CLR     R2                      ;R2 IS TO RECEIVE THE QUOTIENT
(1)  000304'  012001                            MOV     (R0)+,R1                ;THE DIVISOR
(1)  000306'  001422                            BEQ     BCD3                    ;EXIT IF ZERO
(1)  000310'  160103                    BCD5:   SUB     R1,R3                   ;DIVIDE BY SUBTRACTING
(1)  000312'  103402                            BCS     BCD4
(1)  000314'  005202                            INC     R2                      ;UP THE QUOTIENT AFTER EACH SUB
(1)  000316'  000774                            BR      BCD5
(1)  000320'  060103                    BCD4:   ADD     R1,R3                   ;GONE TOO FAR
(1)  000322'  005702                            TST     R2                      ;QUOTIENT =0?
(1)  000324'  001005                            BNE     BCD6                    ;NO
(1)  000326'  005716                            TST     (SP)                    ;LEADING ZERO'S?
(1)  000330'  001003                            BNE     BCD6                    ;NO
(1)  000332'  012702  000040                    MOV     #40,R2                  ;YES PRINT SPACE
(1)  000336'  000403                            BR      BCD7
(1)  000340'  005216                    BCD6:   INC     (SP)                    ;NOMORE LEADING ZERO'S
(1)  000342'  062702  000060                    ADD     #'0,R2                  ;MAKE IT ASCII
(1)  000346'  004767  000776'           BCD7:   JSR     PC,CHROUT               ;PRINT IT
(1)  000352'  000753                            BR      BCD2
(1)  000354'  005726                    BCD3:   TST     (SP)+                   ;POP INDICATOR WORD.
(2)  000356'  000167  002236'                   JMP     RESR5                   ;GO RESTORE REGS 0-4, DO RTS R5.
(1)  000362'  023420  001750  000144            .WORD   10000.,1000.,100.,10.,1
(1)  000370'  000012  000001
(1)  000374'  000000                    DECTAB: .WORD   0
```

```
(1)                                                 .SBTTL  DATE UNPACK AND TYPE SUBROUTINE
(1)   000376' 004467  002040'          DATUPK::     JSR     R4,SAV04                    ;SAVE THEM
(1)   000402' 012704  000105                        MOV     #69.,R4                     ;BASE YEAR IS 1970
(1)   000406' 042703  100000                        BIC     #100000,R3                  ;GET RID OF CONTIG BIT
(1)   000412' 005204                   DATUP1:      INC     R4                          ;SEE!
(1)   000414' 162703  001750                        SUB     #1000.,R3                   ;FIND WHAT YEAR
(1)   000420' 003374                                BGT     DATUP1
(1)   000422' 062703  001750                        ADD     #1000.,R3                   ;WENT TOO FAR
(1)   000426' 012767  000034  000066                MOV     #28.,DATTAB+2               ;ASSUME LEAN YEAR.
(1)   000434' 032704  000003                        BIT     #3,R4                       ;LEAP YEAR?
(1)   000440' 001002                                BNE     DATUP4                      ;BR IF NOT.
(1)   000442' 005267  000054                        INC     DATTAB+2                    ;YES. CORRECT FOR FEB.
(1)   000446' 012700  000520'          DATUP4:      MOV     #DATTAB,R0                  ;GO FIND WHAT MONTH
(1)   000452' 020310                   DATUP3:      CMP     R3,(R0)                     ;LESS THAN WHAT THIS MON HAS
(1)   000454' 003402                                BLE     DATUP2                      ;YES, FOUND THE MONTH
(1)   000456' 162003                                SUB     (R0)+,R3                    ;NO, ADVANCE MONTH
(1)   000460' 000774                                BR      DATUP3
(1)   000462' 004567  177572           DATUP2:      JSR     R5,BCDCV                    ;PRINT OUT THE DAY FIRST
(1)   000466' 000002                                .WORD   2
(1)   000470' 016067  000030  000004                MOV     24.(R0),DATUP5              ;POINT TO MONTH NAME
(1)   000476' 004567  000642'                       JSR     R5,MES                      ;AND PRINT IT
(1)   000502' 000000                   DATUP5:      .WORD   0
(1)   000504' 010403                                MOV     R4,R3                       ;NOW THE YEAR
(1)   000506' 004567  177546                        JSR     R5,BCDCV                    ;PRINT THAT OUT
(1)   000512' 000002                                .WORD   2
(2)   000514' 000167  002022'                       JMP     RESR7                       ;GO RESTORE REGS 0-4, DO RTS PC.
(1)   000520' 000037  000034  000037   DATTAB:      .WORD   31.,28.,31.,30.
(1)   000526' 000036
(1)   000530' 000037  000036  000037                .WORD   31.,30.,31.,31.
(1)   000536' 000037
(1)   000540' 000036  000037  000036                .WORD   30.,31.,30.,31.
(1)   000546' 000037
(1)   000550' 000600'                               $JAN
(1)   000552' 000606'                               $FEB
(1)   000554' 000614'                               $MAR
(1)   000556' 000622'                               $APR
(1)   000560' 000630'                               $MAY
(1)   000562' 000636'                               $JUN
(1)   000564' 000644'                               $JUL
(1)   000566' 000652'                               $AUG
(1)   000570' 000660'                               $SEP
(1)   000572' 000666'                               $OCT
(1)   000574' 000674'                               $NOV
(1)   000576' 000702'                               $DEC
(1)   000600' 045055  047101  000055   $JAN:        .ASCIZ  '-JAN-'
(1)   000606' 043055  041105  000055   $FEB:        .ASCIZ  '-FEB-'
(1)   000614' 046455  051101  000055   $MAR:        .ASCIZ  '-MAR-'
(1)   000622' 040455  051120  000055   $APR:        .ASCIZ  '-APR-'
(1)   000630' 046455  054501  000055   $MAY:        .ASCIZ  '-MAY-'
(1)   000636' 045055  047125  000055   $JUN:        .ASCIZ  '-JUN-'
(1)   000644' 045055  046125  000055   $JUL:        .ASCIZ  '-JUL-'
(1)   000652' 040455  043525  000055   $AUG:        .ASCIZ  '-AUG-'
(1)   000660' 051455  050105  000055   $SEP:        .ASCIZ  '-SEP-'
(1)   000666' 047455  052103  000055   $OCT:        .ASCIZ  '-OCT-'
(1)   000674' 047055  053117  000055   $NOV:        .ASCIZ  '-NOV-'
(1)   000702' 042055  041505  000055   $DEC:        .ASCIZ  '-DEC-'
```

```
(1)                                             .EVEN
```

```
(1)                                             .SBTTL  CHAIN BUFFER
(1)             000000'                         .CSECT  RUNBUF
(1)   000000' 000400                    RUNBUF: .BLKW   256.
(1)                                             .SBTTL  PAK & UNPAK TR BUF
(1)             000000'                         .CSECT  TRBF
```

```
(1)                                         .SBTTL  COMMAND DECODER, INI, AND DELAY ROUTINES
(1)          000000'                    .CSECT  RESMON
(1)                                     ;PROGRAM STACK
(1)  000000' 000024                     R6STCK: .BLKW   20.
(1)  000050'                            SPBOT:
(1)
(1)                                     ;END OF PASS CHAIN MODE ENTRY POINT.
(1)  000050' 005046                     RESTRT::        CLR     -(6)            ;CLEAR T BIT.
(1)  000052' 012746  000060'            $REL11: MOV     #RSTRT1,-(6)            ;WILL RTI TO RSTRT1
(1)  000056' 000002                             RTI                             ;
(1)  000060' 004767  000730             RSTRT1: JSR     PC,CROUT2               ;CHECK FOR CTL C.
(1)  000064' 005737  000042                     TST     @#42                    ;ABORT CURRENT PROGRAM?
(1)  000070' 001425                             BEQ     COMCON                  ;BR IF YES.
(1)  000072' 105767  000121                     TSTB    QVMODE                  ;QUICK VERIFY MODE?
(1)  000076' 001022                             BNE     COMCON                  ;BR IF YES. DO NEXT CHAIN ENTRY.
(1)  000100' 005327                             DEC     (PC)+                   ;ALL PASSES DONE?
(1)  000102' 000000                     PCOUNT: .WORD   0                       ;PASS COUNTER.
(1)  000104' 001417                             BEQ     COMCON                  ;BR IF YES.
(1)  000106' 000207                             RTS     PC                      ;NO. RETURN TO CURRENT PROGRAM.
(1)
(1)                                     ;ERROR REPORTING ROUTINE
(1)  000110' 004767  000066             COMCO1: JSR     PC,DELAY                ;WAIT A BIT.
(2)  000114' 004467  001720                     JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  000120' 010500                             MOV     R5,R0                   ;GET ADDR OF ASCII MESSAGE.
(1)  000122' 004567  000526                     JSR     R5,MESO                 ;TYPE ERROR MESSAGE.
(1)  000126' 105767  004370                     TSTB    RUNID                   ;WAS IT RUN COMMAND?
(1)  000132' 001004                             BNE     COMCON                  ;BR IF YES. CONTINUE CHAIN MODE.
(1)  000134' 012707                             MOV     (PC)+,PC                ;GO TO COMCO3.
(1)  000136' 000140'                    $COMC3: COMCO3
(1)
(1)                                     ;CHAIN MODE IS CLEARED HERE.
(1)  000140' 005027                     COMCO3: CLR     (PC)+                   ;CLEAR CHAIN MODE.
(1)  000142' 000000                     CHN:    .WORD   0                       ;CHAIN MODE INDICATOR.
(1)
(1)                                     ;WHERE EVERYTHING STARTS
(1)  000144' 012706  000050'            COMCON: MOV     #SPBOT,SP               ;SET UP THE STACK
(1)  000150' 004567  001740                     JSR     R5,BCLEAR               ;CLEAR BUFFERS, VARIABLES.
(1)  000154' 003400' 000000  001135     $REL1:  CLRBEG,0,CLREND-CLRBEG
(1)
(1)  000162' 004567  000454             COMCO2: JSR     R5,MES                  ;TYPE A DOT 1ST.
(1)  000166' 004535'                            ADOT
(1)  000170' 004567  000630                     JSR     R5,INPUT                ;GO FETCH A COMMAND
(1)  000174' 004767  000346             COMCO5: JSR     PC,GTCK                 ;CHECK COMMAND SYNTAX
(1)  000200' 000761                             BR      COMCON                  ;TILL EVERYTHING IS DONE
(1)
(1)
(1)  000202' 005046                     DELAY:: CLR     -(SP)                   ;DELAY A LITTLE BIT.
(1)  000204' 005316                     1$:     DEC     (SP)
(1)  000206' 100776                             BMI     1$
(1)  000210' 005726                             TST     (SP)+                   ;RESTORE STACK.
(1)  000212' 000207                     RTSPC:  RTS     PC                      ;DONE. RETURN.
```

```
(1)                                                .SBTTL  CHAIN  SETUP ROUTINE
(1)   000214' 005027                      DOIT:    CLR     (PC)+                       ;AT FIRST BLOCK. ALSO CLEARS QVMODE.
(1)   000216'    000                      BKCT:    .BYTE   0                           ;CONTAINS CHAIN BLOCK #.
(1)   000217'    000                      QVMODE:  .BYTE   0                           ;QUICK VERIFY INDICATOR.
(1)   000220' 012767  041503  004266               MOV     #"CC,IFNAM+6                ;SET UP A CCC EXTENSION.
(1)   000226' 112767  000103  004262               MOVB    #'C,IFNAM+8.
(2)   000234' 004767  001206                       JSR     PC,SETIN                    ;SET INPUT DEVICE. NAME NEEDED.
(1)   000240' 004767  000302                       JSR     PC,GTSW                     ;GET SWITCHES.
(1)   000244' 004567  001634                       JSR     R5,BMOVE                    ;COPY INPUT DDB TO BATCH DDB.
(1)   000250' 003234' 004400' 000117      $REL6:   BTCDDB,INDDB,BTCEND-BTCDDB
(1)   000256' 105267  177734              D01:     INCB    BKCT                        ;WANT FIRST BLOCK OF FILE.
(1)   000262' 010546                      D02:     MOV     R5,-(SP)                    ;SAVE R5.
(1)   000264' 004567  001614                       JSR     R5,BMOVE                    ;BATCH DDB TO INPUT DDB.
(1)   000270' 004400' 003234' 000117      $REL5:   INDDB,BTCDDB,BTCEND-BTCDDB
(2)   000276' 004467  001536                       JSR     R4,SAV04                    ;SAVE REGS 0-4
(1)   000302' 004767  002356                       JSR     PC,INITI                    ;INIT INPUT.
(2)   000306' 004767  001540                       JSR     PC,RST04                    ;RESTORE REGS 0-4
(1)   000312' 116703  177700                       MOVB    BKCT,R3                     ;GET THE REQUIRED BLOCK NUMBER.
(1)   000316' 105067  177674                       CLRB    BKCT                        ;
(2)   000322'                             D03:
(2)   000322' 004767  001370                       JSR     PC,READL                    ;READ LINKED FILE BLOCK.
(1)   000326' 004567  001552                       JSR     R5,BMOVE                    ;INPUT BUFFER TO BATCH BUFFER.
(1)   000332' 000000' 003400' 001000      $REL4:   RUNBUF,BUF,512.
(1)   000340' 012767  000002' 177574      $REL13:  MOV     #RUNBUF+2,CHN               ;SET CHAIN MODE WITH ADDR OF 1ST CHAR.
(1)   000346' 012767  000776  002764               MOV     #510.,RNBK                  ;# OF CHARACTERS IN BUFFER.
(1)   000354' 105267  177636                       INCB    BKCT                        ;INCR # OF BLOCKS READ.
(1)   000360' 005303                               DEC     R3                          ;READ THE WANTED BLOCK?
(1)   000362' 001357                               BNE     D03                         ;BR IF NOT.
(1)   000364' 012605                               MOV     (SP)+,R5                    ;RESTORE R5.
(1)   000366' 005027                      RCKSUM:  CLR     (PC)+                       ;CHECKSUM THE BATCH BUFFER.
(1)   000370' 000000                      RCKSM:   .WORD   0
(1)   000372' 012703  000000'             $REL12:  MOV     #RUNBUF,R3                  ;GET ADDR OF BATCH BUFFER.
(1)   000376' 012704  000400                       MOV     #256.,R4                    ;WILL DO 256 WORDS.
(1)   000402' 062367  177762              1$:      ADD     (R3)+,RCKSM                 ;CHECKSUM A WORD.
(1)   000406' 005304                               DEC     R4                          ;ALL DONE?
(1)   000410' 001374                               BNE     1$                          ;BR IF NOT.
(1)   000412' 000207                               RTS     PC                          ;YES. RETURN.
```

```
(1)                                             .SBTTL  CHAIN EXECUTION ROUTINE
(1)  000414' 020001                     CHAIN:  CMP     R0,R1           ;AT START OF KYBD BUFFER?
(1)  000416' 001021                             BNE     CHAIN0          ;BR IF NOT.
(1)  000420' 016746  177744                     MOV     RCKSM,-(SP)     ;SAVE BATCH CHECKSUM.
(1)  000424' 004767  177736                     JSR     PC,RCKSUM       ;RECHECKSUM THE BUFFER.
(1)  000430' 026726  177734                     CMP     RCKSM,(SP)+     ;MATCH?
(1)  000434' 001412                             BEQ     CHAIN0          ;BR IF YES.
(1)  000436' 016746  177500                     MOV     CHN,-(SP)       ;NO. SAVE CHN
(1)  000442' 016746  002672                     MOV     RNBK,-(SP)      ;SAVE RNBK
(1)  000446' 004767  177610                     JSR     PC,D02          ;GET THE BLOCK.
(1)  000452' 012667  002662                     MOV     (SP)+,RNBK      ;RESTORE RNBK.
(1)  000456' 012667  177460                     MOV     (SP)+,CHN       ;RESTORE CHN.
(1)  000462' 016703  177454             CHAIN0: MOV     CHN,R3          ;NEXT COMMAND
(1)  000466' 016704  002646                     MOV     RNBK,R4         ;COUNT
(1)  000472' 005304                     CHAIN1: DEC     R4
(1)  000474' 100003                             BPL     CHAIN2          ;BR IF POSITIVE
(1)  000476' 004767  177554                     JSR     PC,D01          ;NO. NEED NEW BUFFER.
(1)  000502' 000767                             BR      CHAIN0
(1)  000504' 112302                     CHAIN2: MOVB    (R3)+,R2        ;GET A BYTE
(1)  000506' 001614                             BEQ     COMCO3          ;BACK TO COMCO3.
(1)  000510' 120227  000012                     CMPB    R2,#12          ;LINE FEED?
(1)  000514' 001766                             BEQ     CHAIN1          ;DISREGARD IT.
(1)  000516' 010367  177420                     MOV     R3,CHN          ;SAVE IT
(1)  000522' 010467  002626                     MOV     R4,RNBK         ;THAT TOO
(1)  000526' 000207                             RTS     PC              ;RETURN.
```

```
                                        .SBTTL  COMMAND DECODER
000530' 004767  000654          SETCNT: JSR     PC,ATOI                 ;GET THE COUNT.
000534' 010267  003760                  MOV     R2,ICOUNT               ;STORE IT.
000540' 000402                          BR      GTSW

000542' 105267  177451          SETQV:  INCB    QVMODE                  ;SET QV MODE.

000546'                         GTOK:GTSW:
000546' 012701  004542'                 MOV     #COMTAB,R1              ;DEVICE DECODING COMES HERE
000552' 016702  000074          GTOKK:  MOV     RELCNT,R2               ;GET RELOCATION FACTOR.
000556' 060201                          ADD     R2,R1                   ;CORRECT R1 FOR RELOCATION.

000560' 016700  000340          GTOKX:  MOV     KBPTR,R0                ;GET STRING POINTER
000564' 012104  GTOK1:          MOV     (1)+,R4                 ;GET DISPATCH ADDR.
000566' 012103                          MOV     (1)+,R3                 ;GET POINTER TO NEXT ENTRY.
000570' 122021  GTOK2:          CMPB    (R0)+,(R1)+             ;MATCH THE CHARACTER?
000572' 001007                          BNE     GTOKY
000574' 105711  GTOK3:          TSTB    (R1)                    ;LAST CHR IN ENTRY?
000576' 001374                          BNE     GTOK2                   ;NO,CHECK THE OTHER CHR'S
000600' 010067  000320                  MOV     R0,KBPTR                ;SAVE STRING POINTER.
000604' 060204  GTOK4:          ADD     R2,R4                   ;CORRECT FOR RELOCATION.
000606' 005002                          CLR     R2
000610' 000114                          JMP     (R4)                    ;DISPATCH WHERE NEEDED.

000612' 010301  GTOKY:          MOV     R3,R1                   ;POINT TO NEXT ENTRY.
000614' 060201                          ADD     R2,R1                   ;CORRECT FOR RELOCATION.
000616' 022711  177777                  CMP     #-1,(1)                 ;FILLED?
000622' 001356                          BNE     GTOKX                   ;BR IF YES. KEEP LOOKING.

000624' 004567  177260          INVCMD::        JSR     R5,COMCOI               ;REPORT INVALID COMMAND/S..
000630' 047111  041526  042115           .ASCIZ  'INVCMD/SW'
000636' 051457  000127
                                        .EVEN
```

```
                                        .SBTTL  MESSAGE ROUTINES
000642'                         MES::
000642' 004467  001172                  JSR     R4,SAV04                ;SAVE REGS 0-4
000646' 012500                          MOV     (R5)+,R0                ;MESSAGE BUFFER TO R0
000650' 062700                          ADD     (PC)+,R0                ;CORRECT FOR RELOCATION.
000652' 000000                  RELCNT::        .WORD   0                       ;RELOCATION FACTOR.
000654' 112002                  MES0:   MOVB    (R0)+,R2                ;PICK UP ONE CHR TO R2
000656' 001525                          BEQ     GEX04                   ;BR IF 0.
000660' 004767  000002                  JSR     PC,MES1                 ;GO OUTPUT CHAR.
000664' 000773                          BR      MES0                    ;GO FOR MORE.
000666' 120227  000011          MES1:   CMPB    R2,#11                  ;TAB CODE?
000672' 001427                          BEQ     TAB                     ;BR IF YES TO DO A TAB.
000674' 120227  000045                  CMPB    R2,#'%                  ;%?
000700' 001036                          BNE     CHROUT                  ;BR IF NOT.
                                ;
000702' 010246                  CRLF::  MOV     R2,-(SP)                ;SAVE R2.
000704' 012702  005015                  MOV     #5015,R2                ;OUTPUT CR.
000710' 004767  000062                  JSR     PC,CHROUT
000714' 000302                          SWAB    R2                      ;OUTPUT LINE FEED.
000716' 004767  000054                  JSR     PC,CHROUT
000722' 116702  001462                  MOVB    FILLCT,R2               ;GET READY FOR FILLER CHARS.
000726' 005077  000046          1$:     CLR     @MOUT                   ;FILLER IS 0.
000732' 004767  000044                  JSR     PC,CROUT1
000736' 005302                          DEC     R2                      ;DONE?
000740' 003372                          BGT     1$                      ;BR IF NOT.
000742' 005027                          CLR     (PC)+                   ;CLEAR THE CHAR COUNT.
000744' 000000                  CHRCNT: .WORD   0                       ;CHAR COUNT VARIABLE.
000746' 012602                  CRLF1:  MOV     (SP)+,R2                ;RESTORE R2.
000750' 000207                          RTS     PC                      ;DONE. RETURN.

                                ;TAB SUBROUTINE.
000752' 010246                  TAB::   MOV     R2,-(SP)                ;SAVE R2.
000754' 012702  000040          1$:     MOV     #40,R2                  ;SPACES DO THE TABBING.
000760' 004767  000012                  JSR     PC,CHROUT               ;OUTPUT A SPACE.
000764' 142767  000370  177752          BICB    #370,CHRCNT             ;SEE IF DONE.
000772' 001370                          BNE     1$                      ;BR IF NOT DONE.
000774' 000764                          BR      CRLF1                   ;GO EXIT.

                                ;SUB TO OUTPUT CHARACTER TO CONSOLE OR LINE PRINTER
000776' 110237                  CHROUT::        MOVB    R2,@(PC)+               ;OUTPUT CHAR.
001000' 177566                  MOUT:   .WORD   177566
001002'                         CROUT1:
001002' 105737                          TSTB    @(PC)+                  ;WAIT FOR READY.
001004' 177564                  MREG:   .WORD   177564
001006' 100375                          BPL     CROUT1                  ;BACK IF NOT READY.
001010' 105267  177730                  INCB    CHRCNT                  ;UP CHARACTER COUNT.
001014' 004767  000116          CROUT2: JSR     PC,CKYBD                ;CHECK KEYBOARD.
001020' 000207                          RTS     PC                      ;EXIT.NO CHAR.
001022' 000470                          BR      GETCR1                  ;CHECK FOR CTRL C.
```

```
                                        .SBTTL  INPUT ROUTINE
                                        ;TO CALL 'INPUT' DO A   JSR     R5,INPUT
                                        ;FOLLOWED BY            +       ADR OF MESSAGE TO BE  TYPED PRIOR TO INPUT
001024'                         INPUT:
001024'                         GETIN:
001024' 004467  001010'                 JSR     R4,SAVC4                ;SAVE REGS 0-4
001030' 012700  003354'         GETIN0: MOV     #KBUF,R0                ;INPUT BUFFER
001034' 010001                          MOV     R0,R1                   ;SAVE THE ADDRESS
001036'                         GETI01:
001036' 004767  000120          2$:     JSR     PC,GETCHR               ;GET A CHARACTER
001042' 120227  000141          3$:     CMPB    R2,#141                 ;LESS THAN LOWER CASE A?
001046' 103405                          BLO     1$                      ;BR IF YES.
001050' 120227  000172                  CMPB    R2,#172                 ;HIGHER THAN LOWER CASE Z?
001054' 101002                          BHI     1$                      ;BR IF YES.
001056' 162702  000040                  SUB     #40,R2                  ;MAKE IT UPPER CASE.
001062' 120227  000177          1$:     CMPB    R2,#177                 ;RUBOUT?
001066' 001407                          BEQ     GETI02                  ;YES
001070' 110220                  GETI06: MOVB    R2,(R0)+                ;STORE IT. NOT SPECIAL CHAR.
001072' 120227  000015                  CMPB    R2,#15                  ;CARRIAGE RETURN?
001076' 001407                          BEQ     GETI08                  ;QUITTING TIME
001100' 004767  177562          GETI03: JSR     PC,MES1                 ;ECHO THE CHARACTER
001104' 000754                          BR      GETI01                  ;CONTINUE INPUT
001106' 020100                  GETI02: CMP     R1,R0                   ;RUBOUT, BUFFER EMPTY?
001110' 001752                          BEQ     GETI01                  ;YEP,ECHO CRLF
001112' 114002                  GETI11: MOVB    -(R0),R2                ;GET THE LAST CHR
001114' 000771                          BR      GETI03                  ;AND ECHO IT
001116' 112720  000012          GETI08: MOVB    #12,(R0)+               ;STORE LF TOO.
001122' 010127                          MOV     R1,(PC)+                ;POINT TO START OF KEYBOARD BUFFER.
001124' 000000                  KBPTR:  .WORD   0                       ;KEYBOARD POINTER.
001126' 004767  177550          GEX02:  JSR     PC,CRLF                 ;CRLF.
001132'                         GEX04:
001132' 000167  001100                  JMP     RESR5                   ;GO RESTORE REGS 0-4, DO RTS R5.

001136' 105737                  CKYBD:  TSTB    @(PC)+                  ;KEYBOARD ACTIVE?
001140' 177560                  KTKS:   .WORD   177560
001142' 100006                          BPL     CKYBD1                  ;BR IF NOT.
001144' 013702                          MOV     @(PC)+,R2               ;GET CHARACTER.
001146' 177562                  KTKB:   .WORD   177562
001150' 042702  177600                  BIC     #177600,R2              ;CLEAR OUT JUNK BITS.
001154' 062716  000002                  ADD     #2,(SP)                 ;SET UP CHARACTER IN BUFFER RETURN.
001160' 000207                  CKYBD1: RTS     PC                      ;EXIT.

001162' 005767  176754          GETCHR: TST     CHN                     ;IN CHAIN MODE?
001166' 001403                          BEQ     1$                      ;BR IF NOT.
001170' 004767  177220                  JSR     PC,CHAIN                ;YES. GET CHAR FROM CHAIN FILE.
001174' 000403                          BR      GETCR1
001176' 004767  177734          1$:     JSR     PC,CKYBD                ;WAIT FOR CHARACTER.
001202' 000775                          BR      1$                      ;LOOP TILL YOU GOT ONE.
001204' 120227  000003          GETCR1: CMPB    R2,#3                   ;CTRL C?
001210' 001363                          BNE     CKYBD1                  ;BR IF NOT.
001212' 016707  176720                  MOV     $COMC3,PC               ;YES. TIME TO QUIT.
```

```
(1)                                          .SBTTL  ERROR MESSAGE ROUTINES.
(1)  001216' 004567  176666         DEVERR::        JSR     R5,COMC01       ;REPORT DEVICE ERROR.
(1)  001222' 042504  042526  051122         .ASCIZ  'DEVERR'
(1)  001230'    000
(1)          001232'                        .EVEN
```

```
(1)                                                 .SBTTL  ITOA SUBROUTINE
(1)                                                 ;BINARY TO ASCII ROUTINE
(1)                                                 ;TAKES WHAT'S IN R3 AND SHIFTS THREE BITS INTO R2
(1)                                                 ;THEN CALLS PRINTOUT ROUTINE TO OUTPUT THEM
(1)  001232' 004467  000602          ITOA::  JSR     R4,SAV04                ;I GET SCREWED WHEN I DON'T
(1)  001236' 012704  000006                  MOV     #6,R4                   ;DO ONLY SIX TIMES
(1)  001242' 005002                          CLR     R2                      ;WHERE THE DIGITS GO
(1)  001244' 000413                          BR      ITOA3
(1)  001246' 062702  000060          ITOA1:  ADD     #'0,R2                  ;MAKE IT ASCII
(1)  001252' 004767  177520                  JSR     PC,CHROUT               ;TYPE IT
(1)  001256' 005304                          DEC     R4                      ;ONE DOWN
(1)  001260' 003410                          BLE     ITOA2                   ;IF NOMORE TO GO
(1)  001262' 005002                          CLR     R2                      ;GET RID OF OLD STUFF
(1)  001264' 006303                          ASL     R3                      ;SHIFT COMBINED
(1)  001266' 006102                          ROL     R2                      ;THREE TIMES
(1)  001270' 006303                          ASL     R3                      ;THIS IS
(1)  001272' 006102                          ROL     R2                      ;ACTUALLY FASTER
(1)  001274' 006303                  ITOA3:  ASL     R3                      ;AND MORE EFFICIENT
(1)  001276' 006102                          ROL     R2                      ;THAN A DO LOOP
(1)  001300' 000762                          BR      ITOA1                   ;KEEP GOING
(2)  001302'                         ITOA2:
(2)  001302' 000167  000514                  JMP     RESR7                   ;GO RESTORE REGS 0-4, DO RTS PC.
(1)
```

```
                                        .SBTTL  GETNUM/ATOI SUBROUTINES

001306' 016700  177612          GETNUM: MOV     KBPTR,R0                ;GET STRING POINTER.
001312' 005001                  GTNM1:  CLR     R1                      ;DATA
001314' 005027                          CLR     (PC)+
001316' 000000                  YES:    .WORD   0
001320' 112002                  2$:     MOVB    (R0)+,R2                ;GET A BYTE
001322' 120227  000040                  CMPB    R2,#40                  ;SPACE?
001326' 001771                          BEQ     GTNM1                   ;YES,IGNORE IT
001330' 122702  000015                  CMPB    #15,R2                  ;CR?
001334' 001417                          BEQ     3$                      ;YES,RETURN
001336' 120227  000060                  CMPB    R2,#'0                  ;LOW LIMIT
001342' 002414                          BLT     3$                      ;TOO LOW
001344' 120227  000067                  CMPB    R2,#'7                  ;HIGH LIMIT
001350' 003011                          BGT     3$                      ;TOO HIGH
001352' 006301                          ASL     R1                      ;SHIFT OLD STUFF
001354' 006301                          ASL     R1                      ;3 TIMES LEFT
001356' 006301                          ASL     R1                      ;I.E. MULT. BY OCTAL 10
001360' 060201                          ADD     R2,R1                   ;ADD NEW TO OLD
001362' 162701  000060                  SUB     #'0,R1                  ;BUT GET RID OF ASCII STUFF
001366' 105267  177724                  INCB    YES                     ;SET FLAG
001372' 000752                          BR      2$                      ;MORE, MORE
001374'                         3$:
001374' 005300                  GTNM0:  DEC     R0
001376' 010067  177522                  MOV     R0,KBPTR                ;SAVE STRING POINTER.
001402' 105767  177710          GTNJM1: TSTB    YES
001406' 000207                          RTS     PC                      ;NOMORE

                                ;DECIMAL ASCII TO BINARY CONVERT SUBROUTINE.
001410' 016700  177510          ATOI:   MOV     KBPTR,R0                ;POINT TO STRING
001414' 112003                  ATOI1:  MOVB    (R0)+,R3                ;GET DIGIT.
001416' 162703  000060                  SUB     #'0,R3                  ;CONVERT TO BINARY
001422' 100764                          BMI     GTNM0                   ;NOT A DIGIT.
001424' 020327  000011                  CMP     R3,#9.                  ;CHECK UPPER LIMIT.
001430' 003361                          BGT     GTNM0                   ;TOO HIGH.
001432' 006302                          ASL     R2
001434' 006302                          ASL     R2
001436' 060302                          ADD     R3,R2                   ;ALL DONE.
001440' 000765                          BR      ATOI1
```

```
                                                .SBTTL  DEVICE SETUP ROUTINE, INPUT INIT ROUTINE
001442' 005046                          SETI:   CLR     -(SP)                   ;INDICATE NO NAME.
001444' 000401                                  BR      DVSET
001446' 010746                          SETIN:  MOV     PC,-(SP)                ;INDICATE NAME NEEDED.
001450' 012705                          DVSET:  MOV     (PC)+,R5                ;INPUT DDB ADDR TO R5.
001452' 004462'                         $IDDB:: .WORD   INDEV
001454' 016746  177172                          MOV     RELCNT,-(SP)            ;PUT RELOC FACTOR IN STACK.
001460' 012700  004656'                 $REL16: MOV     #DEVTAB,R0              ;GET DEVICE TABLE ADDR.
001464' 062700                                  ADD     (PC)+,R0                ;ADD OFFSET FOR DESIRED DEVICE.
001466' 000000                          CURDRV: .WORD   0                       ;HOLDS CURRENT DRIVE # TIMES 2.
001470' 011000                                  MOV     (R0),R0                 ;GET DEVICE SET UP ADDR.
001472' 061600                                  ADD     (SP),R0                 ;CORRECT FOR RELOCATION.
001474' 004710                                  JSR     PC,(R0)                 ;GO SET UP DEVICE.
001476' 061600                                  ADD     (SP),R0                 ;CORRECT PARAM ADDR FOR RELOCATION.
001500' 010067  000006                          MOV     R0,$REL7+2
001504' 004567  000374                          JSR     R5,BMOVE                ;MOVE DEVICE INFO TO DDB.
001510' 004436' 000000  000050  $REL7:  INBOOT,OPEN,IFNAM-INBOOT                ;DEST,SOURCE, COUNT.
001516' 016703  177766                          MOV     $REL7,R3
001522' 012701  000011                          MOV     #9.,R1                  ;NUMBER OF ENTRIES TO RELOCATE.
001526' 061623                          5$:     ADD     (SP),(R3)+              ;RELOCATE ENTRY.
001530' 005301                                  DEC     R1                      ;DONE?
001532' 001375                                  BNE     5$                      ;BR IF NOT.
001534' 062665  000020                          ADD     (SP)+,XDR(R5)           ;ANOTHER ONE NEEDS IT.
001540' 005726                                  TST     (SP)+                   ;NAME NEEDED?
001542' 001531                                  BEQ     RESR7A                  ;BR IF NOT.
001544'                                 FILNAM:
001544' 004567  000344                          JSR     R5,BCLEAR               ;CLEAR NAME AREA TO BLANKS.
001550' 004506' 000040  000006  $REL3:  IFNAM,40,6.
001556'                                 FNAM3:
001556' 004467  000256                          JSR     R4,SAV04                ;SAVE REGS 0-4
001562' 016700  177336                          MOV     KBPTR,R0                ;KYBD BUFFER POINTER TO R0.
001566' 012704  004506'                 $REL2:  MOV     #IFNAM,R4
001572' 012703  000006                          MOV     #6,R3                   ;UP TO 6 DIGITS FOR NAME
001576' 112002                          FNAM1:  MOVB    (R0)+,R2                ;GET A CHARACTER.
001600' 120227  000077                          CMPB    R2,#'?                  ;QUESTION MARK?
001604' 001003                                  BNE     1$                      ;BR IF NOT.
001606' 010565  177720                          MOV     R5,XF_MOD(R5)           ;SET FILE MODE INDICATOR.
001612' 000416                                  BR      3$
001614' 120227  000015                  1$:     CMPB    R2,#15                  ;CR?
001620' 001411                                  BEQ     2$                      ;BR IF YES.
001622' 120227  000040                          CMPB    R2,#40                  ;SPACE?
001626' 001763                                  BEQ     FNAM1                   ;BR IF YES.
001630' 120227  000060                          CMPB    R2,#'0                  ;LESS THAN 0?
001634' 103403                                  BLO     2$                      ;BR IF YES. NOT ALPHA-NUMERIC.
001636' 120227  000132                          CMPB    R2,#'Z                  ;HIGHER THAN Z?
001642' 101402                                  BLOS    3$                      ;BR IF NOT. ALPHA-NUMERIC CHAR.
001644' 005300                          2$:     DEC     R0                      ;MOVE POINTER BACK ONE.
001646' 000403                                  BR      5$                      ;GO CLEAN UP.
001650' 110224                          3$:     MOVB    R2,(R4)+                ;STORE THE CHARACTER.
001652' 005303                          4$:     DEC     R3                      ;DONE 6 CHARS?
001654' 003350                                  BGT     FNAM1                   ;BR IF NOT.
001656' 020327  000006                  5$:     CMP     R3,#6                   ;ANY NAME CHARS?
001662' 001403                                  BEQ     INVNAM                  ;BR IF NONE. ERROR.
001664' 010067  177234                          MOV     R0,KBPTR                ;SAVE STRING POINTER.
001670' 000454                                  BR      RESR7                   ;RESTORE REGS, DO RTS PC
```

```
(1)  001672' 004567  176212          INVNAM: JSR     R5,COMC01           ;REPORT INVALID NAME.
(1)  001676' 047111  047126  046501          .ASCIZ  'INVNAM'
(1)  001704'    000
(1)          001706'                         .EVEN
```

```
(1)                                             .SBTTL  READL, READC, AND BKREAD SUBROUTINES.
(1)                                     ;SUB TO READ ONE BLOCK, SET R0 AND R1 POINTERS.
(1)  001706' 016700  000160             GTDATA: MOV     $BUF2,R0                ;ADDR OF 1ST DATA BYTE.
(1)  001712' 012701  000776                     MOV     #510.,R1                ;SET BYTE COUNT IN R1.
(1)
(1)                                     ;SUBROUTINE TO READ LINKED FILE BLOCK INTO BUF
(1)  001716' 016705  177530             READL:  MOV     $IDDB,R5                ;POINT TO INPUT DDB.
(1)  001722' 105267  002575                     INCB    PIPFLG                  ;SET PIP MODE.
(1)  001726' 005765  000006                     TST     XDT(R5)                 ;LAST BLOCK?
(1)  001732' 001436                             BEQ     EOMERR                  ;BR IF YES. ERROR.
(1)  001734' 004767  000024                     JSR     PC,BKREAD               ;READ BLOCK.
(1)  001740' 016765  001434  000006             MOV     BUF,XDT(R5)             ;SAVE NEXT BLOCK ADDRESS.
(1)  001746' 000207             READ_1: RTS     PC                              ;DONE. RETURN.
(1)
(1)                                     ;SUB TO INPUT/OUTPUT NEXT BLOCK.
(1)  001750' 016765  001424  000006     NXTBLK::        MOV     BUF,XDT(R5)             ;GET BLOCK NUMBER.
(1)  001756' 001773                             BEQ     READL1                  ;IF 0. NO MORE. ERROR RETURN.
(1)  001760' 062716  000002                     ADD     #2,(SP)                 ;SET FOR NORMAL RETURN.
(1)
(1)                                     ;SUBROUTINE TO READ A BLOCK INTO BUF.
(1)  001764' 004767  000076             BKREAD::        JSR     PC,CLRBUF       ;CLEAR THE BUFFER 1ST.
(1)  001770' 016765  000106  000004             MOV     $BUF,XBA(R5)            ;SET READ ADDRESS.
(1)  001776' 012765  000400  000002     BKRD0:  MOV     #256.,XWC(R5)           ;SET WORD COUNT.
(1)  002004' 016565  000012  000010     READBK::        MOV     XRD(R5),XCO(R5) ;SET READ COMMAND.
(2)  002012'                            XYBK:
(2)  002012' 004467  000022                     JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  002016' 004775  177774                     JSR     PC,@XSV(R5)             ;DO IT.
(1)
(2)  002022'                            RESR7::
(2)  002022' 004767  000024                     JSR     PC,RST04                ;RESTORE REGS 0-4
(1)  002026' 000207             RESR7A: RTS     PC                              ;RETURN.
(1)
(1)  002030' 004567  176054             EOMERR::        JSR     R5,COMC01       ;REPORT END OF MEDIUM ERROR.
(1)  002034' 047505  000115                     .ASCIZ  'EOM'
(1)                                             .EVEN
```

```
(1)                                             .SBTTL  UTILITY SUBROUTINES
(1)
(1)  002040' 010346                     SAV04:: MOV     R3,-(SP)                ;SAVE R3
(1)  002042' 010246                             MOV     R2,-(SP)                ;SAVE R2
(1)  002044' 010146                             MOV     R1,-(SP)                ;SAVE R1
(1)  002046' 010046                             MOV     R0,-(SP)                ;SAVE R0
(1)  002050' 010407                             MOV     R4,PC                   ;R5 IS ALREADY SAVED
(1)
(1)
(1)  002052' 012604                     RST04:: MOV     (SP)+,R4                ;RETURN ADDRESS
(1)  002054' 012600                             MOV     (SP)+,R0                ;RESTORE R0
(1)  002056' 012601                             MOV     (SP)+,R1                ;R1
(1)  002060' 012602                             MOV     (SP)+,R2                ;R2
(1)  002062' 012603                             MOV     (SP)+,R3
(1)  002064' 000204                             RTS     R4                      ;RESTORE R4 AND RETURN
(1)
```

```
(1)
(1)  002066' 004567  000022          ;SUB TO CLEAR BUFFER.
(1)  002072' 003402' 000000  000776  CLRBUF::       JSR     R5,BCLEAR               ;CALL BYTE CLEAR SUB.
(1)  002100' 000207                  $BUF2:: BUF+2,0,510.                            ;DEST,CLEAR VALUE,COUNT
(1)  002102' 003400'                         RTS     PC                              ;EXIT.
(1)                                  $BUF::  BUF
(1)
(1)                                  ;ROUTINE TO MOVE BYTE FIELDS.
(1)  002104' 012767  112120  000022  BMOVE:: MOV     #112120,BMC2            ;SET A MOVB (1)+,(0)+
(1)  002112' 000403                          BR      BMC1
(1)
(1)                                  ;ROUTINE TO CLEAR BYTE FIELDS TO SPECIFIC VALUE.
(1)  002114' 012767  110120  000012  BCLEAR::        MOV     #110120,BMC2    ;SET A MOVB R1,)0)+
(2)  002122'                         BMC1:
(2)  002122' 004467  177712                  JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  002126' 012500                          MOV     (5)+,R0                 ;GET DEST ADDR.
(1)  002130' 012501                          MOV     (5)+,R1                 ;GET SOURCE.
(1)  002132' 012502                          MOV     (5)+,R2                 ;GET COUNT.
(1   002134' 000000                  BMC2:   OPEN
(1)  002136' 005302                          DEC     R2                      ;DONE?
(1)  002140' 001375                          BNE     BMC2                    ;BR IF NOT.
(1)  002142' 000435                  BMC3:   BR      UPKNM1
(1)
(1)                                  ;CMPNAM SUBROUTINE. COMPARES TWO 9 CHARACTER NAMES. WILD CHARS ALLOWED.
(2)  002144'                         CMPNAM::
(2)  002144' 004467  177670                  JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  002150' 012700  004506'         $REL15: MOV     #IFNAM,R0               ;DESIRED NAME ADDRESS.
(1)  002154' 012701                          MOV     (PC)+,R1                ;ADDR OF NAME UNDER QUESTION.
(1)  002156' 004524'                 $TXNAM::        .WORD   TXNAM
(1)  002160' 012702  000011                  MOV     #9.,R2                  ;COMPARE UP TO 9 CHARACTERS.
(1)  002164'                         1$:
(1)  002164' 122710  000077                  CMPB    #'?,(0)                 ;CHAR A WILD CHARACTER?
(1)  002170' 001002                          BNE     11$                     ;BR IF NOT.
(1)  002172' 122021                          CMPB    (R0)+,(R1)+             ;POINT TO NEXT CHAR.
(1)  002174' 000402                          BR      2$
(1)  002176' 122021                  11$:    CMPB    (0)+,(1)+               ;COMPARE CHARACTERS.
(1)  002200' 001003                          BNE     3$                      ;BR IF NOT SAME.
(1)  002202' 005302                  2$:     DEC     R2                      ;MATCH.DECREMENT COUNT.
(1)  002204' 001367                          BNE     1$                      ;BR IF NOT DONE YET.
(1)  002206' 005725                          TST     (R5)+                   ;DONE. SET UP MATCH EXIT.
(1)  002210' 000412                  3$:     BR      UPKNM1
(1)
(1)                                  ;SUBROUTINE TO CONVERT RAD50 FILE NAME TO ASCII.
(2)  002212'                         UPKNAM::
(2)  002212' 004467  177622                  JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  002216' 012501                          MOV     (R5)+,R1                ;GET ASCII ADDR.
(1)  002220' 012500                          MOV     (R5)+,R0                ;GET RAD50 ADDR.
(2)  002222' 004567  000026                  JSR     R5,UNPACK               ;UNPACK 2 WORDS INTO 6 ASCII BYTES.
(2)  002226' 062700  000004                  ADD     #4,R0                   ;POINT TO EXT ADDR.
(2)  002232' 004567  000006                  JSR     R5,UPACK1               ;UNPACK 1 WORD INTO 3 ASCII BYTES.
(1)  002236'                         JPKNM1:
(2)  002236'                         RESR5::
(2)  002236' 004767  177610                  JSR     PC,RST04                ;RESTORE REGS 0-4
(1)  002242' 000205                          RTS     R5                      ;DONE. RETURN.
```

```
                                                .SBTTL  RAD50 UNPACK SUBROUTINE
                                        ;       INPUT:  R0=ADR OF MOD40 NUMBER (2 WORDS)
                                        ;               R1=ADR OF ASCII STRING (6 BYTES)
                                        ;       OUTPUT: R1 POINTS ONE PAST LAST GENERATED CHARACTER
                                        ;
                                        ;       IF N IS THE MOD40 NUMBER, THEN
                                        ;               N=C1*50↑2+C2*50+C3
                                        ;       THUS, N/50↑2 IS C1 AND THE REMAINDER IS C2*50+C3
                                        ;       THE REMAINDER IS DIVIDED BY 50 TO GET C2 ETC.
(1)  002244' 012727  177777          UPACK1::       MOV     #-1,(PC)+       ;UNPACK ONE WORD ONLY.
(1)  002250' 000000                  PAKTMP: .WORD  0
(1)  002252' 000403                          BR      UNPA07
(1)  002254' 012767  177776  177766 UNPACK::       MOV     #-2,PAKTMP      ;MAJOR LOOP COUNT
(2)  002262'                         UNPA07:
(2)  002262' 004467  177552                  JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  002266' 012704  177775          UNPA09: MOV     #-3,R4          ;MINOR LOOP COUNT
(1)  002272' 011000                          MOV     (R0),R0         ;GET MOD40 WORD
(1)  002274' 012702  002400'         $REL14: MOV     #COEFF,R2       ;PTR TO COEFFICIENT TABLE
(1)  002300' 005003                  UNPA06: CLR     R3              ;0 QUOTIENT
(1)                                  ;       DIVIDE BY COEFFICIENTS
(1)  002302' 020012                  UNPA02: CMP     R0,(R2)         ;DONE WITH DIVIDE
(1)  002304' 103403                          BLO     UNPA01          ;YES
(1)  002306' 161200                          SUB     (R2),R0         ;NO-SUBTRACT COEFF.
(1)  002310' 005203                          INC     R3              ;ADD 1 TO QUOTIENT
(1)  002312' 000773                          BR      UNPA02
(1)                                  ;       DIVIDE DONE.  QUOT IN R3, REMAINDER IN R0
(1)                                  ;       CONVERT TO AN ASCII CHARACTER
(1)  002314' 105703                  UNPA01: TSTB    R3
(1)  002316' 001406                          BEQ     UNPA03          ;"BLANK"
(1)  002320' 120327  000033                  CMPB    R3,#33
(1)  002324' 001407                          BEQ     UNPA05          ;"$"
(1)  002326' 003004                          BGT     UNPA04          ;"." OR "0-9"
(1)  002330' 062703  000040                  ADD     #40,R3          ;"A-Z"
(1)  002334' 062703  000016          UNPA03: ADD     #16,R3
(1)  002340' 062703  000011          UNPA04: ADD     #11,R3
(1)  002344' 062703  000011          UNPA05: ADD     #11,R3
(1)  002350' 110321                          MOVB    R3,(R1)+        ;STORE CHARACTER
(1)  002352' 005722                          TST     (R2)+           ;ADVANCE TO NEXT COEFF.
(1)  002354' 005204                          INC     R4              ;DONE 3 CHARS?
(1)  002356' 002750                          BLT     UNPA06          ;NO-DO MORE
(1)  002360' 011600                          MOV     (SP),R0         ;RESTORE ORIGINAL R0 AND
(1)  002362' 005720                          TST     (R0)+           ;MOVE TO NEXT WORD
(1)  002364' 005267  177660                  INC     PAKTMP          ;DONE 2 WORDS
(1)  002370' 002736                          BLT     UNPA09          ;NO
(1)                                  ;       DONE--PUT CURRENT R1 ONTO THE STACK
(1)  002372' 010166  000002          UNPA08: MOV     R1,2(SP)
(1)  002376' 000717                          BR      UPKNM1          ;GO EXIT.
(1)                                  ;
(1)  002400' 003100  000050  000031  COEFF:  .WORD   1600.,40.,1.     ;40.**2, 40.**1,40.**0
```

F03


```
(1)                                          .SBTTL  FILL, RUN, AND START ROUTINES
(1)  002406' 012703                   FILL:   MOV     (PC)+,R3                   ;GET READY TO TYPE FILL COUNT.
(1)  002410' 000014                   FILLCT: .WORD   14
(1)  002412' 004767  176614                   JSR     PC,ITOA                    ;PRINT THAT OUT
(1)  002416' 004767  176330                   JSR     PC,TAB                     ;TAB OVER.
(1)  002422' 004567  176376                   JSR     R5,INPUT                   ;WAIT FOR INPUT
(1)  002426' 004767  176654                   JSR     PC,GETNUM                  ;CONVERT INPUT STRING TO BINARY
(1)  002432' 001402                           BEQ     MOD1                       ;JUST A CR,DO NOTHING
(1)  002434' 010167  177750                   MOV     R1,FILLCT                  ;PUT WHAT HE ENTERED THERE
(1)  002440' 000207                   MOD1:   RTS     PC
(1)
(1)                                   ;START ROUTINE
(1)  002442' 004767  176540           START:  JSR     PC,GETNUM                  ;FETCH STARTING ADR
(1)  002446' 001420                           BEQ     RUN10                      ;BR IF NO DATA TYPED.
(1)  002450' 006201                           ASR     R1                         ;GOOD ADDRESS?
(1)  002452' 103404                           BCS     INVADR                     ;BR IF NOT (ODD).
(1)  002454' 060101                           ADD     R1,R1                      ;RESTORE ADDR.
(1)  002456' 010167  000454                   MOV     R1,STADR                   ;SAVE ADDR.
(1)  002462' 000412                           BR      RUN10                      ;DATA TYPED.
(1)  002464' 004567  175420           INVADR: JSR     R5,COMC01                  ;REPORT INVALID ADDRESS.
(1)  002470' 047111  040526  051104           .ASCIZ  'INVADR'
(1)  002476'    000
(1)          002500'                          .EVEN
(1)
(1)  002500' 105267  002016           RUN:    INCB    RUNID                      ;SET RUN INDICATOR.
(1)  002504' 004767  000216                   JSR     PC,LOAD                    ;DO A LOAD 1ST.
(1)  002510' 004767  176032           RUN10:  JSR     PC,GTSW                    ;GET SWITCHES.
(1)  002514' 016767  002000  175360           MOV     ICOUNT,PCOUNT              ;GET I COUNT IF ANY.
(1)  002522' 001002                           BNE     RUN11                      ;BR IF NON ZERO.
(1)  002524' 005267  175352                   INC     PCOUNT                     ;MAKE IT ONE.
(1)  002530' 005767  175406           RUN11:  TST     CHN                        ;CHAIN MODE?
(1)  002534' 001403                           BEQ     RUN20                      ;BR IF NOT.
(1)  002536' 012737  000050' 000042   $REL10: MOV     #RESTRT,@#42               ;SET RESTART ADDR IN LOC 42.
(1)  002544' 016701  000366           RUN20:  MOV     STADR,R1                   ;
(1)  002550' 006201                           ASR     R1                         ;
(1)  002552' 103002                           BCC     RUN30                      ;BR IF EVEN ADDR.
(1)  002554' 012701  000100                   MOV     #100,R1                    ;ODD ADDR START AT 200.
(1)  002560' 006301                   RUN30:  ASL     R1                         ;RESTORE THE ADDR.
(1)  002562' 016746  177622           RUN40:  MOV     FILLCT,-(SP)               ;PASS THE FILL COUNT.
(1)  002566' 012746  012345                   MOV     #12345,-(SP)               ;INDICATE XXDP MONITOR LOAD.
(1)  002572' 000111                           JMP     (R1)                       ;START THE PROGRAM.
(1)
```

```
                                        .SBTTL  DIR ROUTINE
002574' 004767  176642          DIR:    JSR     PC,SETI                 ;NO NAME NEEDED.
002600' 012746                  DIR:    MOV     (PC)+,-(SP)             ;GET FILE POSITION COUNT.
002602' 000001                  FILCNT: .WORD   1
002604' 004767  0000006                 JSR     PC,CREWND               ;REWIND TAPE.
002610' 004767  0000006         1$:     JSR     PC,CRDHDR               ;READ FILE LABEL.
002614' 000436                          BR      FLNOTF                  ;EOT RETURN.
002616' 005316                          DEC     (SP)                    ;GOT TO FILE?
002620' 001373                          BNE     1$                      ;BR IF NOT.
002622' 005726                          TST     (SP)+                   ;RESTORE STACK.
002624' 012765  010000  000002          MOV     #MONCNT,XWC(R5)         ;4K'S WORTH. STARTING AT LOC 0
002632' 005065  000004                  CLR     XBA(R5)                 ;XFR STARTS AT 0.
002636' 004767  177142                  JSR     PC,READBK               ;DO IT.
002642' 012701  000000'                 MOV     #NRDIR,R1               ;POINT TO NON-RES DIR ROUTINE.
002646' 016746  176252                  MOV     KBPTR,-(SP)             ;PASS THE BUFFER POINTER.
002652' 016746  175260                  MOV     $COMC3,-(SP)            ;PASS MONITOR RESTART ADDR.
002656' 016746  176604                  MOV     CURDRV,-(SP)            ;PASS CURRENT DRIVE.
002662' 000737                          BR      RUN40                   ;GO TO NON-RESIDENT DIR ROUTINE.
                                                                        ;VIA RUN40.

                                ;INPUT INIT ROUTINE
002664' 016705  176562          INITI:  MOV     $IDDB,R5                ;POINT TO INPUT DDB.
002670' 010565  177744                  MOV     R5,XISTBK(R5)           ;DUMMY BLOCK NUMBER.
002674' 004775  177770                  JSR     PC,@SRH(R5)             ;FILE SEARCH.
002700' 000404                          BR      2$                      ;FILE NOT FOUND.
002702' 016565  177744  000006          MOV     XISTBK(R5),XDT(R5)      ;1ST BLOCK # TO INDT
002710' 000207                          RTS     PC                      ;FOUND. RETURN.
002712'                         2$:
002712' 004567  175172          FLNOTF::        JSR     R5,COMCO1       ;REPORT FILE NOT FOUND.
002716' 042516  043130  046111           .ASCIZ  'NEXFIL'
002724'     000
        002726'                         .EVEN
```

```
                                        .SBTTL  LOAD ROUTINE. ;.BIN OR .BIC FILES ONLY
002726'                         LOAD:
002726' 004767  176514                  JSR     PC,SETIN                ;SET INPUT DEVICE. NAME NEEDED.
002732' 012767  044502  001554          MOV     #"BI,IFNAM+6            ;SET UP BIC EXTENSION.
002740' 112767  000103  001550          MOVB    #'C,IFNAM+8.
002746' 005767  175170                  TST     CHN                     ;IN CHAIN MODE?
002752' 001005                          BNE     1$                      ;BR IF YES.
002754' 112767  000077  001534          MOVB    #'?,IFNAM+8.            ;NO. MAKE LAST CHAR WILD.
002762' 010565  177720                  MOV     R5,XFLMOD(R5)           ;INDICATE FILE MODE.
002766'                         1$:
002766' 004767  177672                  JSR     PC,INITI                ;INIT FOR INPUT.
002772' 004767  176710          LOAD1:  JSR     PC,GTDATA               ;INPUT A BLOCK OF DATA
002776' 005027                  LOAD2:  CLR     (PC)+                   ;INITIALIZE CHECKSUM
003000' 000000                  CHKSUM: .WORD   0
003002' 004767  000154                  JSR     PC,RDFRAM               ;READ A SYNC WORD
003006' 005703                          TST     R3                      ;GOT A NULL?
003010' 001772                          BEQ     LOAD2                   ;IF YES, KEEP READING.
003012' 005303                          DEC     R3                      ;SEE IF IT'S A ONE
003014' 001025                          BNE     CKSMER                  ;IF NOT, LOAD ERROR.
003016' 004767  000140                  JSR     PC,RDFRAM               ;SYNC IS A WORD
003022' 105703                          TSTB    R3                      ;OF 1
003024' 001021                          BNE     CKSMER                  ;SO THE SECOND HALF MUST BE 0
003026' 004767  000154                  JSR     PC,RD2FRM               ;2 BYTES=1 WORD
003032' 010304                          MOV     R3,R4                   ;ASSUMING NOT DONE YET
003034' 162704  000004                  SUB     #4,R4                   ;MINUS THE HEADER
003040' 022704  000002                  CMP     #2,R4                   ;BYTE COUNT=6?
003044' 001431                          BEQ     LJMP                    ;IT IS, THE END IS NEAR
003046' 004767  000134                  JSR     PC,RD2FRM               ;GET LOAD ADR
003052' 010302                          MOV     R3,R2                   ;INTO R2
003054' 004767  000102          LOAD3:  JSR     PC,RDFRAM               ;GET A BYTE
003060' 100016                          BPL     LOAD4                   ;BYTE COUNT NOT ZERO YET
003062' 105767  177712                  TSTB    CHKSUM                  ;CHECK SUM SHOULD BE ZERO
003066' 001743                          BEQ     LOAD2                   ;IT IS
003070' 004567  175014          CKSMER: JSR     R5,COMCOI               ;REPORT LOAD ERROR.
003074' 045503  046523  051105          .ASCIZ  'CKSMER'
003102'     000
                003104'                 .EVEN
003104' 004567  175000          POFLOW: JSR     R5,COMCOI               ;PROGRAM OVERFLOW MESSAGE.
003110' 047520  046106  000117          .ASCIZ  'POFLO'
                                        .EVEN
003116' 020227  000000'         LOAD4:  CMP     R2,#R6STCK              ;PROTECT THE MONITOR
003122' 103370                          BHIS    POFLOW                  ;ABORT
003124' 110322                          MOVB    R3,(R2)+                ;STORE THE BYTE.
003126' 000752                          BR      LOAD3                   ;GO GET MORE
003130' 004767  000052          LJMP:   JSR     PC,RD2FRM               ;GET THE JUMP ADR
003134' 010327                          MOV     R3,(PC)+                ;STORE IT FOR RAINY DAYS
003136' 000001                  STADR:  .WORD   1
003140' 004767  000016                  JSR     PC,RDFRAM               ;MAKE SURE THE CHECKSUM IS OK
003144' 105767  177630                  TSTB    CHKSUM                  ;WE CHECK EVERY BLOCK
003150' 001347                          BNE     CKSMER
003152' 112737  000004  000041          MOVB    #4,@#41         ;SET LOAD MEDIUM INDICATOR.
003160' 000207                          RTS     PC                      ;DONE. GET OUT.
003162' 005301                  RDFRAM: DEC     R1                      ;BYTE COUNT IN BUFFER
003164' 100003                          BPL     RDFRMA                  ;SOMETHING IN BUFFER
003166' 004767  176514                  JSR     PC,GTDATA               ;NO. GET ANOTHER BUFFER FULL
003172' 000773                          BR      RDFRAM                  ;DO THE HOUSE KEEPING
```

```
                003174' 112003          RDFRMA: MOVB    (R0)+,R3                ;PICK UP CHR
                003176' 060367  177576          ADD     R3,CHKSUM               ;DO THE CHECKSUM STUFF
                003202' 005304                  DEC     R4                      ;LOAD BYTE COUNT
                003204' 000207          RDFRMB: RTS     PC
                003206' 004767  177750  RD2FRM: JSR     PC,RDFRAM               ;GET ONE BYTE FIRST
                003212' 010327                  MOV     R3,(PC)+                ;STORE IT TEMPORARILY
                003214' 000000          LTEMP:  .WORD   0                       ;TEMP STORAGE.
                003216' 004767  177740          JSR     PC,RDFRAM               ;GET THE OTHER BYTE
                003222' 110367  177767          MOVB    R3,LTEMP+1              ;INTO THE HIGH BYTE
                003226' 016703  177752          MOV     LTEMP,R3                ;BACK INTO R3
                003232' 000207                  RTS     PC                      ;RETURN
```

```
                                        .SBTTL  BATCH DEVICE DESCRIPTOR BLOCK (DDB)
003234'                                 BTCDDB:
003234'  000000                         RWCTR:  0
003236'  000000                         RFLMOD: 0
003240'  000000                         RFLCNT: 0
003242'  000000                         RSVMAP: 0
003244'  000000                         RSVCNT: 0
003246'  000000                         RSVBLK: 0
003250'  000000  000000                 RSVNAM: 0,0
003254'  000000                         RSVEXT: 0
003256'  000000                         RSVDAT: 0
003260'  000000                         RSVXX:  0
003262'  000000                         R1STBK: 0
003264'  000000                         RBKLGT: 0
003266'  000000                         RLSTBK: 0
003270'  000000                         RSVUPT: 0
003272'  000000                         RBOOT:  0
003274'  000000                         RDRT:   0
003276'  000000                         RZER:   0
003300'  000000                         RDLT:   0
003302'  000000                         RCLS:   0
003304'  000000                         RRETR:  0
003306'  000000                         RSRCH:  0
003310'  000000                         RALC:   0
003312'  000000                         RSRV:   0
003314'  000000                         RDRV:   0
003316'  000000                         RCM:    0
003320'  000000                         RWC:    0
003322'  000000                         RBA:    0
003324'  000000                         RDT:    0
003326'  000000                         RCOM:   0
003330'  000000                         RPRC:   0
003332'  000000                         RRPWC:  0
003334'  000000                         RRBKCT: 0
003336'  000000                         RDIR:   0
003340'  000000                         RNBK:   0
003342'     000      000      000       RFNAM:  .BYTE   0,0,0,0,0,0,0,0,0
003345'     000      000      000
003350'     000      000      000
003353'                                 BTCEND:
            003354'                             .EVEN
```

```
(1)
(1)  003354' 000012                    ;KEYBOARD BUFFER.
(1)                                    KBUF:   .BLKW   10.
(1)
(1)
(1)                                            .SBTTL  START OF CLEARABLE CORE (DURING INIT)
(1)  003400'                           CLRBEG:                 ;BEGINNING OF CLEARABLE AREA (DURING INIT).
(1)                                    ;MAIN READ - WRITE BUFFER
(1)  003400' 000400                    BUF::   .BLKW   256.
```

```
                                        .SBTTL  INPUT DEVICE DESCRIPTOR BLOCK (DDB)
004400'                         DDBSTR:INDDB:
004400' 000000                  IWCTR:  .WORD   0       ;XWCTR    OUTPUT FILE OPEN FLAG
004402' 000000                  IFLMOD::        .WORD   0       ;XFLMOD   FILE MODE FLAG
004404' 000000                  IFLCNT: .WORD   0       ;XFLCNT   FILE COUNT
004406' 000000                  ISVMAP: .WORD   0       ;XSVMAP   BLK # OF FILE'S MAP BLOCK
004410' 000000                  ISVCNT: .WORD   0       ;XSVCNT   ENTRY # OF CURR FILE IN UFD
004412' 000000                  ISVBLK: .WORD   0       ;XSVBLK   BLK # OF CURR FILE'S UFD

                                ;UFD DIRECTORY ENTRY DATA FOR FILE
004414' 000000  000000          ISVNAM: .WORD   0,0     ;XSVNAM   FILE'S FILENAME IN RAD50 (2 WORDS)
004420' 000000                  ISVEXT: .WORD   0       ;XSVEXT   FILE'S EXTENSION IN RAD50
004422' 000000                  ISVDAT: .WORD   0       ;XSVDAT   FILE'S CREATION DATE IN DOS FORMAT
004424' 000000                  ISVXX:  .WORD   0       ;XSVXX    (NOT USED?)
004426' 000000                  IISTBK: .WORD   0       ;XISTBK   BLOCK # OF FILE'S FIRST DATA BLK
004430' 000000                  IBKLGT: .WORD   0       ;XBKLGT   # OF BLOCKS IN THE FILE
004432' 000000                  ILSTBK: .WORD   0       ;XLSTBK   BLOCK # OF LAST DATA BLOCK WRITTEN
004434' 000000                  ISVUPT: .WORD   0       ;XSVUPT   (NOT USED?)
                                ;END OF DIRECTORY ENTRIES

004436' 000000                  INBOOT: .WORD   0       ;XBT      ADDRESS OF "BOOT" ROUTINE
004440' 000000                  INDRT:  .WORD   0       ;DRT      ADDRESS OF "DIRECTORY" ROUTINE
004442' 000000                  INZER:  .WORD   0       ;ZER      ADDRESS OF "ZERO" ROUTINE
004444' 000000                  INDLT:  .WORD   0       ;DLT      ADDRESS OF "DELETE" ROUTINE
004446' 000000                  INCLS:  .WORD   0       ;CLS      ADDRESS OF "CLOSE" ROUTINE
004450' 000000                  INETR:  .WORD   0       ;ETR      ADDRESS OF "ENTER" (CREATE) ROUTINE
004452' 000000                  INSRH:  .WORD   0       ;SRH      ADDRESS OF "LOOKUP" (SEARCH) ROUTINE
004454' 000000                  INALC:  .WORD   0       ;ALC      ADDRESS OF "ALLOCATE" ROUTINE
004456' 000000                  INSRV:  .WORD   0       ;XSV      ADDRESS OF DEVICE DRIVER ROUTINE
004460' 000000                  INDRV:  .WORD   0       ;XDN      CURRENT DRIVE (UNIT) NUMBER
004462'                         INDEV:                  ;R5 POINTS HERE
004462' 000000                  INCM:   .WORD   0       ;XCM      ADDRESS OF DEVICE'S COMMAND REGISTER
004464' 000000                  INWC:   .WORD   0       ;XWC      CURRENT WORD COUNT
004466' 000000                  INBA:   .WORD   0       ;XBA      CURRENT BUS (MEMORY) ADDRESS
004470'                         INBLK:
004470' 000000                  INDT:   .WORD   0       ;XDT      CURRENT BLOCK NUMBER
004472' 000000                  INCOM:  .WORD   0       ;XCO      CURRENT COMMAND CODE
004474' 000000                  INPRC:  .WORD   0       ;XRD      READ COMMAND CODE
004476' 000000                  INPWC:  .WORD   0       ;XWT      WRITE COMMAND CODE
004500' 000000                  INBKCT: .WORD   0       ;XBC      REQUESTED BLOCK COUNT
004502' 000000                  INDIR:  .WORD   0       ;XDR      ADDRESS OF FIRST DIRECTORY BLK #
004504' 000000                  INNBK:  .WORD   0       ;XNB      LAST BLOCK # ALLOCATED (NEXT BLK #)
004506'    000      000     000 IFNAM:  .BYTE   0,0,0,0,0,0,0,0,0       ;XXNAM    FILE'S NAME IN ASCII (9 CHAR'S)
004511'    000      000     000
004514'    000      000     000
           004520'                              .EVEN
004520'                         DDBEND:
```

```
(1)                                             .SBTTL  INITIALIZABLE VARIABLES/ASCII STRINGS
(1)
(1)  004520' 000000                     ICOUNT: .WORD   0
(1)  004522'    000                     RUNID:  .BYTE   0
(1)  004523'    000                     PIPFLG::        .BYTE   0
(1)  004524'    000     000     000     TXNAM:: .BYTE   0,0,0,0,0,0
(1)  004527'    000     000     000
(1)  004532'    000     000     000     TXEXT:  .BYTE   0,0,0
(1)  004535'                            CLREND:
(1)
(1)                                             :ASCII STRINGS
(1)  004535'    045  000056             ADOT:   .ASCIZ  '%.'
(1)  004540'    011     000             ATAB:   .BYTE   11,0
(1)                                             .EVEN
```

```
                                        .SBTTL  COMMAND, SWITCH, AND DEVICE TABLES
                                        ;ALL COMMANDS ARE CHECKED AGAINST THE QUOTES
                                        ;NO ABBREVIATIONS ALLOWED
004542'                         COMTAB:
004542' 000542'                         .WORD   SETQV                   ;DISPATCH ADDRESS FOR /QV
004552' 000530'                         .WORD   SETCNT                  ;DISPATCH ADDRESS FOR /
004560' 000144'                         .WORD   COMCON                  ;DISPATCH ADDRESS FOR .
004566' 002406'                         .WORD   FILL                    ;DISPATCH ADDRESS FOR F
004574' 002442'                         .WORD   START                   ;DISPATCH ADDRESS FOR S
004602' 002726'                         .WORD   LOAD                    ;DISPATCH ADDRESS FOR L
004610' 002500'                         .WORD   RUN                     ;DISPATCH ADDRESS FOR R
004616' 000214'                         .WORD   DOIT                    ;DISPATCH ADDRESS FOR C
004624' 002574'                         .WORD   DIR                     ;DISPATCH ADDRESS FOR D
004632' 000212'                         .WORD   RTSPC                   ;DISPATCH ADDRESS FOR E
004640' 000212'                         .WORD   RTSPC                   ;DISPATCH ADDRESS FOR <15>
004646' 000546'                         .WORD   GTCK                    ;DISPATCH ADDRESS FOR <40>
004654' 177777                          .WORD   -1


004656'                         DEVTAB:
004656' 0000003                         .WORD   SETMTO
                                ;END OF SOURCE.
000001                                  .END
```

```
ADDT    004535R     005  1333#
AK      001411          1333#*
ALC   = 177772          1333#
ALTMOD= 000033          1333#
ALT1  = 000033          1333#
ALT2  = 000175          1333#
ALT3  = 000176          1333#
ARSTRT  001416          1333#
ATAB    004540R     005  1333#
ATOI    001410R     005  1333#
ATOI1   001414R     005  1333#
BCDCV   000260R G       1333#
BCD1    000274R         1333#
BCD2    000302R         1333#
BCD3    000354R         1333#
BCD4    000320R         1333#
BCD5    000310R         1333#
BCD6    000340R         1333#
BCD7    000346R         1333#
BCLEAR  002114R G   005  1333#
BEGIN   001000          1333#
BKCT    000216R     005  1333#*
BKRDO   001776R     005  1333#
BkREAD  001764R G   005  1333#
BMC1    002122R     005  1333#
BMC2    002134R     005  1333#*
BMC3    002142R     005  1333#
BMOVE   002104R G   005  1333#
BTCDDB  003234R     005  1333#
BTCEND  003353R     005  1333#
BUF     003400R G   005  1333#
BURST   000130          1333#
CHAIN   000414R     005  1333#
CHAINO  000462R     005  1333#
CHAIN1  000472R     005  1333#
CHAIN2  000504R     005  1333#
CHKSUM  003000R     005  1333#*
CHN     000142R     005  1333#*
CHRCNT  000744R     005  1333#*
CHROUT  000776R G   005  1333#
CKSMER  003070R     005  1333#
CKYBD   001136R     005  1333#
CKYBD1  001160R     005  1333#
CLRBEG  003400R     005  1333#
CLRBUF  002066R G   005  1333#
CLREND  004535R     005  1333#
CLS   = 177764          1333#
CMPNAM  002144R G   005  1333#
COEFF   002400R     005  1333#
COMCON  000144R     005  1333#
COMC01  000110R     005  1333#
COMC02  000162R     005  1333#
COMC03  000140R     005  1333#
COMC05  000174R     005  1333#
COMTAB  004542R     005  1333#
CRDHDR= ******  G       1333#
```

```
CREWND= ******  G          1333#
CRLF    000702R G    005   1333#
CRLF1   000746R      005   1333#
CROUT1  001002R      005   1333#
CROUT2  001014R      005   1333#
CURDRV  001466R      005   1333#*
DATTAB  000520R            1333#*
DATUPK  000376R G          1333#
DATUP1  000412R            1333#
DATUP2  000462R            1333#
DATUP3  000452R            1333#
DATUP4  000446R            1333#
DATUP5  000502R            1333#*
DDBEND  004520R      005   1333#
DDBSTR  004400R      005   1333#
DECTAB  000374R            1333#
DELAY   000202R G    005   1333#
DEVERR  001216R G    005   1333#
DEVTAB  004656R      005   1333#
DIR     002574R      005   1333#
DLT   = 177762             1333#
DOIT    000214R      005   1333#
DO1     000256R      005   1333#
DO2     000262R      005   1333#
DO3     000322R      005   1333#
DRT   = 177756             1333#
DVSET   001450R      005   1333#
EOMERR  002030R G    005   1333#
ETR   = 177766             1333#
FHELP   001460             1333#
FILCNT  002602R      005   1333#
FILL    002406R      005   1333#
FILLCT  002410R      005   1333#*
FILNAM  001544R      005   1333#
FLNOTF  002712R G    005   1333#
FNAM1   001576R      005   1333#
FNAM3   001556R      005   1333#
FSTMOD  000064R G          1333#*
GETCHR  001162R      005   1333#
GETCR1  001204R      005   1333#
GETIN   001024R      005   1333#
GETINO  001030R      005   1333#
GETIO1  001036R      005   1333#
GETIO2  001106R      005   1333#
GETIO3  001100R      005   1333#
GETIO6  001070R      005   1333#
GETIO8  001116R      005   1333#
GETIII  001112R      005   1333#
GETNUM  001306R      005   1333#
GEXO2   001126R      005   1333#
GEXO4   001132R      005   1333#
GTDATA  001706R      005   1333#
GTNMO   001374R      005   1333#
GTNM1   001312R      005   1333#
GTNUM1  001402R      005   1333#
GTOK    000546R      005   1333#
```

```
GTOKK   000552R     005  1333#
GTOKX   000560R     005  1333#
GTOKY   000612R     005  1333#
GTOK1   000564R     005  1333#
GTOK2   000570R     005  1333#
GTOK3   000574R     005  1333#
GTOK4   000604R     005  1333#
GTSW    000546R     005  1333#
IBKLGT  004430R     005  1333#
ICOUNT  004520R     005  1333#*
IFLCNT  004404R     005  1333#
IFLMOD  004402R G   005  1333#
IFNAM   004506R     005  1333#*
ILSTBK  004432R     005  1333#
INALC   004454R     005  1333#
INBA    004466R     005  1333#
INBKCT  004500R     005  1333#
INBLK   004470R     005  1333#
INBOOT  004436R     005  1333#
INCLS   004446R     005  1333#
INCM    004462R     005  1333#
INCOM   004472R     005  1333#
INDDB   004400R     005  1333#
INDEV   004462R     005  1333#
INDIR   004502R     005  1333#
INDLT   004444R     005  1333#
INDRT   004440R     005  1333#
INDRV   004460R     005  1333#
INDT    004470R     005  1333#
INETR   004450R     005  1333#
INIT    000070          1333#
INITI   002664R     005  1333#
INNBK   004504R     005  1333#
INPRC   004474R     005  1333#
INPUT   001024R     005  1333#
INPWC   004476R     005  1333#
INSRH   004452R     005  1333#
INSRV   004456R     005  1333#
INVADR  002464R     005  1333#
INVCMD  000624R G   005  1333#
INVNAM  001672R     005  1333#
INWC    004464R     005  1333#
INZER   004442R     005  1333#
ISVBLK  004412R     005  1333#
ISVCNT  004410R     005  1333#
ISVDAT  004422R     005  1333#
ISVEXT  004420R     005  1333#
ISVMAP  004406R     005  1333#
ISVNAM  004414R     005  1333#
ISVUPT  004434R     005  1333#
ISVXX   004424R     005  1333#
ITOA    001232R G   005  1333#
ITOA1   001246R     005  1333#
ITOA2   001302R     005  1333#
ITOA3   001274R     005  1333#
IWCTR   004400R     005  1333#
```

```
I1STBK  004426R      005  1333#
KBPTR   001124R      005  1333#*
KBUF    003354R      005  1333#
KCODE   001236            1333#
KTKB    001146R      005  1333#
KTKS    001140R      005  1333#
LIMIT   001232            1333#
LJMP    003130R      005  1333#
LOAD    002726R      005  1333#
LOAD1   002772R      005  1333#
LOAD2   002776R      005  1333#
LOAD3   003054R      005  1333#
LOAD4   003116R      005  1333#
LPB     000142R           1333#
LPS     000140R           1333#
LPSW    000104R           1333#
LTEMP   003214R      005  1333#*
MES     000642R G    005  1333#
MESO    000654R      005  1333#
MESI    000666R      005  1333#
MNBK  = 000024            1333#
MNINST  001547            1333#
MOD1    002440R      005  1333#
MONCNT= 010000            1333#
MOUT    001000R      005  1333#*
MREG    001004R      005  1333#*
MTBA  = 164006            1333#*
MTCM  = 164000            1333#*
MTST  = 164002            1333#*
MTWC  = 164004            1333#*
NAME    001330            1333#
NOCORE  001435            1333#
NRDIR   000000R           1333#
NRGTSW  000074R           1333#
NRSWTB  000144R           1333#
NXTBLK  001750R G    005  1333#
OPEN  = 000000            1333#
PACK    000250            1333#
PAKTMP  002250R      005  1333#*
PC    =%000007            1333#*
PCOUNT  000102R      005  1333#*
PIPFLG  004523R G    005  1333#*
POFLOW  003104R      005  1333#
PRTY7 = 000340            1333#
PS    = 177776            1333#
PSW   = 177776            1333#
QVMODE  000217R      005  1333#*
RALC    003310R      005  1333#
RBA     003322R      005  1333#
RBKLGT  003264R      005  1333#
RBOOT   003272R      005  1333#
RCKSM   000370R      005  1333#*
RCKSUM  000366R      005  1333#
RCLS    003302R      005  1333#
RCM     003316R      005  1333#
RCOM    003326R      005  1333#
```

```
RDFRAM  003162R      005  1333#
RDFRMA  003174R      005  1333#
RDFRMB  003204R      005  1333#
RDIR    003336R      005  1333#
RDLT    003300R      005  1333#
RDRT    003274R      005  1333#
RDRV    003314R      005  1333#
RDT     003324R      005  1333#
RD2FRM  003206R      005  1333#
READBK  002004R G    005  1333#
READL   001716R      005  1333#
READL1  001746R      005  1333#
READY   000262            1333#
REED    000176            1333#
RELCNT  000652R G    005  1333#*
RESR5   002236R G    005  1333#
RESR7   002022R G    005  1333#
RESR7A  002026R      005  1333#
RESTRT  000050R G    005  1333#
RFLCNT  003240R      005  1333#
RF_MOD  003236R      005  1333#
RFNAM   003342R      005  1333#
RLSTBK  003266R      005  1333#
RNBK    003340R      005  1333#*
RPRC    003330R      005  1333#
RRBKCT  003334R      005  1333#
RRETR   003304R      005  1333#
RRPWC   003332R      005  1333#
RSRCH   003306R      005  1333#
RSRV    003312R      005  1333#
RSTRT1  000060R      005  1333#
RST04   002052R G    005  1333#
RSVBLK  003246R      005  1333#
RSVCNT  003244R      005  1333#
RSVDAT  003256R      005  1333#
RSVEXT  003254R      005  1333#
RSVMAP  003242R      005  1333#
RSVNAM  003250R      005  1333#
RSVUPT  003270R      005  1333#
RSVXX   003260R      005  1333#
RTN     000310            1333#
RTSPC   000212R      005  1333#
RUN     002500R      005  1333#
RUNBUF  000000R      003  1333#
RUNID   004522R      005  1333#*
RUN10   002510R      005  1333#
RUN11   002530R      005  1333#
RUN20   002544R      005  1333#
RUN30   002560R      005  1333#
RUN40   002562R      005  1333#
RWC     003320R      005  1333#
RWCTR   003234R      005  1333#
RWND    000112            1333#
RZER    003276R      005  1333#
R0    =%000000            1333#*
R1    =%000001            1333#*
```

```
R1STBK  003262R      005  1333#
R2     =%000002           1333#*
R3     =%000003           1333#*
R4     =%000004           1333#*
R5     =%000005           1333#*
R6     =%000006           1333#*
R6STCK  000000R      005  1333#
SAVO4   002040R G    005  1333#
SETCNT  000530R      005  1333#
SETFST  000066R           1333#
SETI    001442R      005  1333#
SETIN   001446R      005  1333#
SETMTO= ******  G          1333#
SETQV   000542R      005  1333#
SP     =%000006           1333#*
SPBOT   000050R      005  1333#
SR4    = 177770           1333#
STADR   003136R      005  1333#*
START   002442R      005  1333#
TAB     000752R G    005  1333#
TADP   = ****** U          1333
TAPERR  000306             1333#
TRDP   = 000000            1253     1333
TXEXT   004532R      005  1333#
TXNAM   004524R G    005  1333#
UNPACK  002254R G    005  1333#
UNPA01  002314R      005  1333#
UNPA02  002302R      005  1333#
UNPA03  002334R      005  1333#
UNPA04  002340R      005  1333#
UNPA05  002344R      005  1333#
UNPA06  002300R      005  1333#
UNPA07  002262R      005  1333#
UNPA08  002372R      005  1333#
UNPA09  002266R      005  1333#
UPACK1  002244R G    005  1333#
UPKNAM  002212R G    005  1333#
UPKNM1  002236R      005  1333#
XBA    = 000004           1333#*
XBC    = 000016           1333#
XBKLGT= 177746            1333#
XBT    = 177754           1333#
XCM    = 000000           1333#
XCO    = 000010           1333#*
XDN    = 177776           1333#
XDR    = 000020           1333#*
XDT    = 000006           1333#*
XFLCNT= 177722            1333#
XFLMOD= 177720            1333#*
XLSTBK= 177750            1333#
XNB    = 000022           1333#
XRD    = 000012           1333#
XSV    = 177774           1333#
XSVBLK= 177730            1333#
XSVCNT= 177726            1333#
XSVCAT= 177740            1333#
```

```
XSVEXT= 177736            1333#
XSVMAP= 177724            1333#
XSVNAM= 177732            1333#
XSVUPT= 177752            1333#
XSVXX = 177742            1333#
XWC   = 000002            1333#*
XWCTR = 177716            1333#
XWT   = 000014            1333#
XXNAM = 000024            1333#
XYBK    002012R     005   1333#
XYSTBK= 177744            1333#*
YES     001316R     005   1333#*
ZER   = 177760            1333#
$APR    000622R           1333#
$AUG    000652R           1333#
$BUF    002102R G   005   1333#
$BUF2   002072R G   005   1333#
$CCMC3  000136R     005   1333#*
$DEC    000702R           1333#
$FEB    000606R           1333#
$IODB   001452R G   005   1333#
$JAN    000600R           1333#
$JUL    000644R           1333#
$JUN    000636R           1333#
$LITTB  002624            1333#
$MAR    000614R           1333#
$MAY    000630R           1333#
$NOV    000674R           1333#
$OCT    000666R           1333#
$REL1   000154R     005   1333#
$REL10  002536R     005   1333#
$REL11  000052R     005   1333#
$REL12  000372R     005   1333#
$REL13  000340R     005   1333#
$REL14  002274R     005   1333#
$REL15  002150R     005   1333#
$REL16  001460R     005   1333#
$REL2   001566R     005   1333#
$REL3   001550R     005   1333#
$REL4   000332R     005   1333#
$REL5   000270R     005   1333#
$REL6   000250R     005   1333#
$REL7   001510R     005   1333#*
$SEP    000660R           1333#
$TPNM1  000202R G         1333#
$TXNAM  002156R G   005   1333#
.     = 004660R     005   1333#
```

INITI	41#	1333
READL	50#	1333
RESREG	26#	1333
RESR5	29#	1333
RESR7	32#	1333
ROUTIN	53#	
SAVREG	23#	1333
SETBN	47#	
SETIN	44#	1333
TOKN	5#	1333
TOKNS	14#	1333
UNPACK	35#	1333
UPACK1	38#	1333
XDPMON	1109#	1333
[illegible]	1254#	1333
[illegible]	63#	1333

```
ADD       1333
ASL       1333
ASR       1333
BCC       1333
BCS       1333
BEQ       1333
BGE       1333
BGT       1333
BHI       1333
BHIS      1333
BIC       1333
BICB      1333
BIS       1333
BIT       1333
BLE       1333
BLO       1333
BLOS      1333
BLT       1333
BMI       1333
BNE       1333
BPL       1333
BR        1333
CLR       1333
CLRB      1333
CMP       1333
CMPB      1333
DEC       1333
HALT      1333
INC       1333
INCB      1333
JMP       1333
JSR       1333
MOV       1333
MOVB      1333
NOP       1333
ROL       1333
RTI       1333
RTS       1333
SUB       1333
SWAB      1333
TST       1333
TSTB      1333
.ASCII    1333
.ASCIZ    1333
.ASECT    1333
.BLKW     1333
.BYTE     1333
.CSECT    1333
.END      1335
.ENDC     1332    1333
.EVEN     1333
.GLOBL    1333
.IFDF     1253    1333
.IFF      1333
.IFT      1333
.IIF      1333
```

```
TRDP - XXDP TR79F MONITOR M-11-DMQUF-A  MACY11 27(732)  03-JAN-77  13:49  PAGE 4-1
DMQUFA.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                    SEQ 0049

.LIMIT   1333
.LIST       3       4    1333
.MACRO      5      14      23      25      29      32      35      38      41      44      47      50      53      63     109
         1254
.NLIST      1       2    1333
.PAGE    1333
.REM     1333
.SBTTL   1333
.TITLE   1333
.WORD    1333


. ABS.  002710     000
        000710     001
MTDIRT  000000     002
RUNBUF  001000     003
TRBF    000000     004
RESMON  004660     005


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *TRDPM,TRDPM.LST/CRF=DMQUFA.P11/EQ:TRDP
 RUN-TIME: 11 16 2 SECONDS
 RUN-TIME RATIO: 93/30=3.1
 CORE USED:  16K  (31 PAGES)
```

```
                        .LIST   ME
                        .NLIST  MC,MD,CND,TOC
                        .TITLE  TR79F -  XXDP TR79F MODULE READ-ONLY
                        .SBTTL  EQUATES
000000                  R0      =%0
000001                  R1      =%1
000002                  R2      =%2
000003                  R3      =%3
000004                  R4      =%4
000005                  R5      =%5
000006                  R6      =%6
000006                  SP      =%6
000007                  PC      =%7
177776                  PS      =177776
000000                  OPEN    =0
000200                  DONE    =200
100000                  ERROR   =100000
177716                  XWCTR   =-62    ;INDEX TO WRITE COUNTER.
177720                  XFLMOD  =-60    ;INDEX TO FILE MODE INDICATOR
177722                  XFLCNT  =-56    ;INDEX TO FILE COUNT
177724                  XSVMAP  =-54    ;
177726                  XSVCNT  =-52    ;
177730                  XSVBLK  =-50    ;
177732                  XSVNAM  =-46    ;PHONY UFD BLOCK POINTERS
177736                  XSVEXT  =-42    ;
177740                  XSVDAT  =-40    ;
177742                  XSVXX   =-36    ;
177744                  XISTBK  =-34    ;
177746                  XBKLGT  =-32    ;
177750                  XLSTBK  =-30    ;
177752                  XSVUPT  =-26    ;
177754                  XBT     =-24    ;INDEX TO BOOT ROUTINE.
177756                  DRT     =-22    ;INDEX TO DIRECTORY ROUTINE
177760                  ZER     =-20    ;INDEX TO ZERO ROUTINE
177762                  DLT     =-16    ;INDEX TO DELETE ROUTINE
177764                  CLS     =-14    ;INDEX TO CLOSE ROUTINE
177766                  ETR     =-12    ;INDEX TO ENTER ROUTINE
177770                  SRH     =-10    ;INDEX TO LOKUP ROUTINE
177772                  ALC     =-6     ;INDEX TO ALLOCATE ROUTINE
177774                  XSV     =-4     ;INDEX TO SERVICE ROUTINE (DRIVER).
177776                  XDN     =-2     ;DRIVE NUMBER INDEX
000000                  XCM     =0      ;INDEX TO COMMAND REGISTER
000002                  XWC     =2      ;INDEX TO WORD COUNT
000004                  XBA     =4      ;INDEX TO BUS ADDRESS
000006                  XDT     =6      ;INDEX TO BLOCK NUMBER
000010                  XCO     =10     ;INDEX TO COMMAND
000012                  XRD     =12     ;INDEX TO READ COMMAND
000014                  XWT     =14     ;INDEX TO WRITE COMMAND
000016                  XBC     =16     ;INDEX TO REQUESTED BLOCK COUNT
000020                  XDR     =20     ;INDEX TO 1ST DIR BLOCK POINTER.
000022                  XNB     =22     ;INDEX TO LAST BLOCK # ALLOCATED.
000024                  XXNAM   =24     ;INDEX TO ASCII NAME IN DDB
```

```
(1)                                     .SBTTL  GLOBAL REFERENCE DEFINITIONS
(1)                                 ;EXTERNAL GLOBAL DEFINITIONS
(1)
(1)                                     .GLOBL  BMOVE,BCLEAR,CHROUT,CR.LF
(1)                                     .GLOBL  CLRBUF,CMPNAM
(1)                                     .GLOBL  DEVERR,ITOA,MES
(1)                                     .GLOBL  NXTBLK,TXNAM,RELCNT
(1)                                     .GLOBL  FLNOTF,READBK
(1)                                     .GLOBL  RST04,SAV04,UNPACK,UPACK1
(1)                                     .GLOBL  EOMERR,BKREAD
(1)                                     .GLOBL  BUF,DELAY
(1)                                     .GLOBL  PIPFLG,UPKNAM,IFLMOD
(1)                                     .GLOBL  $TXNAM,$BUF2,$BUF
(1)                                     .GLOBL  RESR7,INVCMD
(1)                                     .GLOBL  BCDCV
(1)                                     .GLOBL  DATUPK
(1)                                     .GLOBL  TAB
(1)                                     .GLOBL  $TPNM1
(1)                                     .GLOBL  FSTMOD
```

```
1001
 (1)  000000' 000000G          PARAM:  INVCMD
 (1)  000002' 000000'                  DIRECT
 (1)  000004' 000000G                  INVCMD
 (1)  000006' 000000G                  INVCMD
 (1)  000010' 000000G                  INVCMD
 (1)  000012' 000000G                  INVCMD
 (1)  000014' 000040'                  LOOKUP
 (1)  000016' 000000G                  INVCMD
 (1)  000020' 000560'                  DRIVER
 (1)  000022' 000000           UNIT:   0                       ;UNIT #
 (1)  000024' 164000           CMDREG: 164000                  ;COMMAND REGISTER ADDR
 (1)  000026' 000000           WCOUNT: 0                       ;WORD COUNT
 (1)  000030' 000000           BUSADR: 0                       ;BUS ADDRESS
 (1)  000032' 000000           BLOCK:  0                       ;BLOCK NUMBER
 (1)  000034' 000000           COMD:   0                       ;COMMAND
 (1)  000036' 000005           READ:   5                       ;READ COMMAND
 (1)  000040' 000003           WRITE:  3                       ;WRITE COMMAND
 (1)  000042' 000000           RBKCT:  0                       ;REQUESTED BLOCK COUNT
 (1)  000044' 000050'          DIRPTR: DIRBLK                  ;POINTS TO 1ST DIR BLOCK.
 (1)  000046' 000000           LSTBLK: 0                       ;LAST BLOCK # ALLOCATED
 (1)  000050'                  PAREND:
 (1)
1002
 (1)                           ;PARAMETER TABLE FOR MTCOM SECTION.
 (1)  000050' 177777           DIRBLK: -1                      ;-1 INDICATES SEQUENTIAL FILE DEVICE.
 (1)  000052' 000106'                  REWIND                  ;POINTS TO REWIND ROUTINE.
 (1)  000054' 000136'                  SKIPR                   ;POINTS TO SKIP REVERSE BLOCK ROUTINE.
 (1)  000056' 000000G                  INVCMD                  ;CLRHDR POINTER.
 (1)  000060' 000000G                  INVCMD                  ;WRTHDR POINTER.
 (1)  000062' 000000G                  INVCMD                  ;WRTEOF POINTER.
 (1)  000064' 000342'                  RDHDR                   ;POINTS TO RDHDR ROUTINE.
 (1)  000066' 000000G                  INVCMD                  ;WEOT POINTER
 (1)
1003  000070' 164002           MTST:   164002
1004  000072' 164000           MTCM:   164000
1005  000074' 164004           MTWC:   164004
1006  000076' 164006           MTBA:   164006
1007
```

```
1009
1010  000100'                          SETMTO::  .SBTTL  SETMT ROUTINE
1012  000100' 012700  000000'                   MOV     #PARAM,R0               ;POINT TO PARAM TABLE.
1013  000104' 000207                            RTS     PC                      ;DONE. RETURN.
1014
```

```
1016                                                    .SBTTL  MAGTAPE ROUTINES
1017  000106' 004767  000052          REWIND: JSR     PC,MTDRV                ;SELECT DRIVE
1018  000112' 032777  000040  177750          BIT     #40,@MTST               ;AT LOAD PIONT
1019  000120' 001003                          BNE     1$                      ;BRANCH IF YES
1020  000122' 052703  000021                  BIS     #21,R3                  ;REWIND
1021  000126' 000436                          BR      MTDO                    ;CHCK FOR DONE
1022  000130' 004767  000022          1$:     JSR     PC,WRTIDB               ;GO WRITE IDB
1023  000134' 000207                          RTS     PC                      ;RETURN
1024  000136' 004767  000022          SKIPR:  JSR     PC,MTDRV                ;SELECT DRIVE
1025  000142' 052703  000011                  BIS     #11,R3                  ;BACK SPACE
1026  000146' 012765  000001  000002          MOV     #1,XWC(R5)
1027  000154' 000423                          BR      MTDO
1028  000156' 012703  000005          WRTIDB: MOV     #5,R3           ;W/R IDB
1029  000162' 000420                          BR      MTDO            ;
1030  000164' 005003                  MTDRV:  CLR     R3
1031  000166' 116503  000013                  MOVB    XRD+1(R5),R3            ;GET DRIVE NUMBER
1032  000172' 000303                          SWAB    R3                      ;PUT IN LEFT HALF
1033  000174' 010377  177672                  MOV     R3,@MTCM                ;SELECT THE DRIVE.
1034  000200' 004767  000066                  JSR     PC,MTDO2                ;WAIT FOR DRIVE RDY
1035  000204' 000207                          RTS     PC                      ;RETURN.
1051
1052
1053  000206' 012767  000002' 000000' TRBSET: MOV     #TRBUF,TRBF             ;
1054  000214' 066767  000000G 000000'         ADD     RELCNT,TRBF                     ;
1055  000222' 000207                          RTS     PC                      ;
1056
1057
1058
1059
1060  000224' 016577  000002  177642  MTDO:   MOV     XWC(R5),@MTWC           ;GET WORD COUNT
1061  000232' 005477  177636                  NEG     @MTWC
1062  000236' 004767  177744                  JSR     PC,TRBSET               ;
1063  000242' 016777  000000' 177626          MOV     TRBF,@MTBA              ;MEM. ADDR.
1069  000250' 005077  177614          1$:     CLR     @MTST                   ;CLEAR STATUS
1070  000254' 010377  177612                  MOV     R3,@MTCM                ;SET THE FUNCTION AND GO
1071  000260' 032777  100200  177604  MTDO1:  BIT     #ERROR+DONE,@MTCM       ;DONE?
1072  000266' 001774                          BEQ     MTDO1                   ;WAITING
1073  000270' 100413                          BMI     MTERR                   ;ERROR
1074  000272' 032777  000001  177570  MTDO2:  BIT     #1,@MTST                ;DRIVE READY?
1075  000300' 001774                          BEQ     MTDO2                   ;LOOP TILL DONE
1076  000302' 032777  000040  177560          BIT     #40,@MTST               ;LOAD PIONT
1077  000310' 001402                          BEQ     1$                      ;
1078  000312' 004767  177640                  JSR     PC,WRTIDB               ;
1079  000316' 000207                  1$:     RTS     PC
1080  000320' 032777  000010  177542  MTERR:  BIT     #10,@MTST               ;EOF?
1081  000326' 001403                          BEQ     MTER1                   ;NO
1082  000330' 105267  000000'                 INCB    EOFSEN                  ;SET EOF SEEN FLAG
1083  000334' 000756                          BR      MTDO2                   ;WAIT FOR DRIVE TO SETTLE DOWN
1084  000336' 000167  000000G         MTER1:  JMP     DEVERR                  ;DEVICE ERROR.
1085
1086
1087
1088
1089
1090
1091
```

```
                                ;1092
                                ;1093
                                ;1094
                                ;1095
                                ;1096
                                ;1097
                                ;1098
                                ;1099
                                ;1100
                                ;1101
                                ;1102
001342' 105067  000000'         RDHDR:  CLRB    EOFSEN                  ;KEEP TRACK OF EOF'S SEEN
001346' 004767  177612          RHDR5:  JSR     PC,MTDRV                ;SEE WHAT HAPPENS
001352' 012767  177762  000176        MOV     #-14.,BYTCNT            ;SET BYTE COUNT
001360' 012777  177762  177506        MOV     #-14.,@MTWC
001366' 016767  000000G 000160        MOV     $BUF,DES
001374' 004767  177606                JSR     PC,TRBSET               ;
001400' 016777  000000' 177470        MOV     TRBF,@MTBA
001406' 005077  177456                CLR     @MTST                   ;CLEAR INHIBIT
001412' 016577  000012  177452        MOV     XRD(R5),@MTCM           ;DO A READ
001420' 004767  177646                JSR     PC,MTDO2                ;TEMP FIX MAKE SURE OF DONE
001424' 032777  100200  177440  RHDR1: BIT     #ERROR+DONE,@MTCM       ;DONE?
001432' 001774                        BEQ     RHDR1                   ;NO
001434' 100422                        BMI     RHDR3                   ;ERROR
001436' 032777  000010  177424  RHDR2: BIT     #10,@MTST               ;EOF?
001444' 001407                        BEQ     RHDR6                   ;NO
001446' 105267  000000'               INCB    EOFSEN                  ;UP COUNT OF EOF'S SEEN.
001452' 022767  000003  000000'       CMP     #3,EOFSEN               ;SEEN 3?
001460' 001405                        BEQ     RHDR4                   ;BR IF YES. EOT.
001462' 000731                        BR      RHDR5
001464' 062716  000002          RHDR6: ADD     #2,@SP                  ;NO ERRORS SKIP RETURN
001470' 004767  000020                JSR     PC,PAK                  ;GO PACK THE BUFFER
001474' 005767  000000G         RHDR4: TST     BUF                     ;
001500' 000207                        RTS     PC
001502' 032777  001000  177360  RHDR3: BIT     #1000,@MTST             ;BLOCK LENGTH ERROR?
001510' 001316                        BNE     RHDR5                   ;DATA BLOCK
001512' 000711                        BR      MTER1                   ;SOME KINDA ERROR.
                                ;1129
                                ;1130
                                ;1131
                                ;1132
                                ;1133
                                ;1134
000514'                         PAK:
 (1) 000514' 004467  000000G           JSR     R4,SAV04                ;SAVE REGS 0-4
1136 000520' 004767  177462            JSR     PC,TRBSET               ;
1137 000524' 016700  000000'           MOV     TRBF,R0         ;GET INPUT BUFFER
1138 000530' 016701  000020            MOV     DES,R1                  ;GET OUTPUT BUFFER
1139 000534' 112021                1$: MOVB    (R0)+,(R1)+             ;PACK IT IN
1140 000536' 005200                    INC     R0                      ;POINT TO NEXT WORD
1141 000540' 005267  000012            INC     BYTCNT                  ;COUNT THE BYTE
1142 000544' 001373                    BNE     1$                      ;BR IF NOT FINISHED
1143 000546' 004767  000000G           JSR     PC,RST04                ;RESTORE REGS 0-4
1144 000552' 000207                    RTS     PC
                                ;1145
                                ;1146
```

```
000554' 000000                          DES:     000
000556' 000000                          BYTCNT:  000

                                        ;THE DRIVER
000560' 105067  000000'                 DRIVER:  CLRB    EOFSEN                 ;SAW NO EOF YET
000564' 016567  000004  177762                   MOV     XBA(R5),DES            ;LOAD POINTER
000572' 012767  177000  177756          1$:      MOV     #-512.,BYTCNT          ;256 WORDS PER RECORD
000600' 012777  177000  177266          11$:     MOV     #-512.,@MTWC           ;SET WORD COUNT IN DEVICE
000606' 004767  177374                           JSR     PC,TRBSET              ;
000612' 016777  000000' 177256                   MOV     TRBF,@MTBA
000620' 004767  177340                           JSR     PC,MTDRV               ;SELECT THE DRIVE.
000624' 005077  177240                           CLR     @MTST                  ;CLEAR INHIBIT
000630' 016577  000010  177234                   MOV     XCO(R5),@MTCM          ;EXECUTE COMMAND
000636' 004767  177416                           JSR     PC,MTDO1               ;WAIT FOR DONE
000642' 004767  177646                           JSR     PC,PAK
000646' 062767  001000  177700          2$:      ADD     #512.,DES              ;UPDATE
000654' 105767  000000'                          TSTB    EOFSEN                 ;EOF ???
000660' 001007                                   BNE     3$                     ;BRANCH IF YES
000662' 105767  000000G                          TSTB    PIPFLG                 ;PIP MODE?
000666' 001004                                   BNE     3$                     ;YES, EXIT
000670' 162765  000400  000022                   SUB     #256.,XWC(R5)          ;DECREMENT WC
000676' 003335                                   BGT     1$                     ;NOT DONE YET
000700' 000207                          3$:      RTS     PC
```

```
1222
1223
1224
1225
1226
1227
1228
```

```
                        000000'                         .CSECT  MTCOMM
                                                :       TR79F MAGTAPE FILE HEADER FORMAT
                                                :
                                                :       FILE HEADER CONSISTS OF SEVEN (7) WORDS.
                                                :
                                                :       WORD 1 THROUGH 3                NAME AND EXTENSION IN RAD5C
                                                :
                                                :       WORD 4                          PPN (401)
                                                :
                                                :       WORD 5                          0
                                                :
                                                :       WORD 6                          DATE (DOS FORMAT). IF CONTIGUOUS
                                                :                                       FILE, BIT 15 IS SET TO 1.
                                                :
                                                :       WORD 7                          FILE SIZE IN # OF 256 WORD BLOCKS.
 (1)  000000'    000     000            EOFSEN: .BYTE   0,0
                                                .SBTTL  DISPATCH TO UNIQUE DEVICE ROUTINES.

                        000000                  CDIR=   0
                        000002                  CRWD=   2
                        000004                  CSKPR=  4
                        000006                  CCLHD=  6
                        000010                  CWTHD=  10
                        000012                  CWEOF=  12
                        000014                  CRHD=   14

 (1)  000002'                           CREWND::
 (1)  000002' 012746  000002                    MOV     #CRWD,-(SP)             ;POINT TO REWIND ROUTINE.
 (1)  000006' 000405                            BR      COMMON

 (1)  000010' 012746  000004           CSKIPR:  MOV     #CSKPR,-(SP)            ;POINT TO SKIPR ROUTINE.
 (1)  000014' 000402                            BR      COMMON

 (1)  000016'                           CRDHDR::
 (1)  000016' 012746  000014                    MOV     #CRHD,-(SP)             ;POINT TO RDHDR ROUTINE.

 (1)  000022' 066516  000020           COMMON:  ADD     XDR(R5),(SP)            ;ADD ADDR OF ROUTINE TABLE.
 (1)  000026' 017616  000000                    MOV     @(SP),(SP)              ;GET ADDR OF DESIRED ROUTINE.
 (1)  000032' 066716  000006                    ADD     RELCNT,(SP)             ;CORRECT FOR RELOCATION.
 (1)  000036' 012667                            MOV     (SP)+,PC                ;GO TO DESIRED ROUTINE.
```

```
                                                .SBTTL  LOOKUP ROUTINE
                                ;
                                ;
000040' 005765  177722          LOOKUP: TST     XFLCNT(R5)              ;FILE COUNT 0?
000044' 001403                          BEQ     1$                      ;BR IF YES. GO REWIND.
000046' 005765  177720                  TST     XFLMOD(R5)              ;IN FILE MODE?
000052' 001002                          BNE     SEARCH                  ;BR IF YES.
000054' 004767  177722          1$:     JSR     PC,CREWND               ;REWIND TAPE.
000060' 004767  177732          SEARCH: JSR     PC,CRDHDR               ;READ HEADER BLOCK.
000064' 000443                          BR      4$                      ;NOT FOUND. AT EOT.
000066' 005265  177722                  INC     XFLCNT(R5)              ;INCREMENT FILE COUNT.
000072' 005767  000000G                 TST     BUF                     ;DELETED FILE?
000076' 001770                          BEQ     SEARCH                  ;BR IF YES. SKIP IT.
000100' 016767  000000G 000012          MOV     $TXNAM,2$
000106' 016767  000000G 000006          MOV     $BUF,2$+2
000114' 004567  000000G                 JSR     R5,UPKNAM               ;CONVERT NAME TO ASCII.
000120' 000000  000000          2$:     OPEN,OPEN
000124' 004567  000000G                 JSR     R5,CMPNAM               ;COMPARE NAMES.
000130' 000753                          BR      SEARCH                  ;NO MATCH.
000132' 016765  000000G 177732          MOV     BUF,XSVNAM(R5)          ;STORE NAME IN DDB.
000140' 016765  000002G 177734          MOV     BUF+2,XSVNAM+2(R5)
000146' 016765  000004G 177736          MOV     BUF+4,XSVEXT(R5)        ;STORE EXTENSION.
000154' 016765  000012G 177740          MOV     BUF+10.,XSVDAT(R5)      ;THE DATE TOO.
000162' 016765  000014G 177746          MOV     BUF+12.,XBKLGT(R5)      ;FILE LENGTH ALSO.
000170' 062716  000002                  ADD     #2,(SP)                 ;SET UP SUCCESS EXIT.
000174' 000207                  4$:     RTS     PC                      ;DONE. RETURN.
                                ;
                                ;
                                ;
                                ;
```

```
(1)          000000'                   .CSECT  TRBF
(1) 000000'  000002'          TRBF:    TRBUF
(1) 000002'  001000           TRBUF:   .BLKW   512.
(1)                                    .SBTTL  DIRECTORY ROUTINE
(1)          000000'                   .CSECT  MTDIRT
(1) 000000'  004567  000000G  DIRECT:  JSR     R5,MES            ;WHAT KINDA TAPE
(1) 000004'  000074'                   HEADER
(1) 000006'  004767  000040'  1$:      JSR     PC,LOOKUP         ;FIND A FILE.
(1) 000012'  000426                    BR      2$                ;BR IF NO MORE.
(1) 000014'  004767  000000G           JSR     PC,CRLF           ;CRLF
(1) 000020'  016503  177722            MOV     XFLCNT(R5),R3     ;TYPE FILE COUNT.
(1) 000024'  004767  000000G           JSR     PC,ITOA           ;DO IT.
(1) 000030'  004767  000000G           JSR     PC,TAB            ;TAB.
(1) 000034'  004767  000000G           JSR     PC,$TPNM1         ;TYPE FILE NAME.
(1) 000040'  016503  177740            MOV     XSVDAT(R5),R3     ;TYPE FILE DATE.
(1) 000044'  004767  000000G           JSR     PC,DATUPK         ;DO IT.
(1) 000050'  005765  177740            TST     XSVDAT(R5)        ;CONTIGUOUS FILE?
(1) 000054'  100354                    BPL     1$                ;BR IF NOT.
(1) 000056'  112702  000103            MOVB    #'C,R2            ;YES. TYPE A C.
(1) 000062'  004767  000000G           JSR     PC,CHROUT         ;DO IT.
(1) 000066'  000747                    BR      1$                ;GO FOR MORE.
(1) 000070'  000167  000002'  2$:      JMP     CREWND            ;REWIND TAPE
(1) 000074'                   HEADER:
(1) 000074'  042445  052116  054522    .ASCIZ  '%ENTRY#'<11>'FILNAM.EXT'<11>'DATE%'
(1) 000102'  004443  044506  047114
(1) 000110'  046501  042456  052130
(1) 000116'  042011  052101  022505
(1) 000124'     000
(1)          000126'                   .EVEN
2308         000001                    .END
```

```
ALC    = 177772            1000#
BCDCV  = ******  G         1000#
BCLEAR= ******  G          1000#
BKREAD= ******  G          1000#
BLOCK   000032R            1001#
BMOVE  = ******  G         1000#
BUF    = ******  G         1000#   1124    1229
BUSADR  000030R            1001#
BYTCNT  000556R            1105*   1141*   1190#   1195*
CCLHD  = 000006            1229#
CDIR   = 000000            1229#
CHROUT= ******  G          1000#   1229
CLRBUF= ******  G          1000#
CLS    = 177764            1000#
CMDREG  000024R            1001#
CMPNAM= ******  G          1000#   1229
COMD    000034R            1001#
COMMON  000022R     002    1229#
CRDHDR  000016R G   002    1229#
CREWND  000002R G   002    1229#
CRHD   = 000014            1229#
CRLF   = ******  G         1000#   1229
CRWD   = 000002            1229#
CSKIPR  000010R     002    1229#
CSKPR  = 000004            1229#
CWEOF  = 000012            1229#
CWTHD  = 000010            1229#
DATUPK= ******  G          1000#   1229
DELAY  = ******  G         1000#
DES     000554R            1107*   1138    1189#   1194*   1213*
DEVERR= ******  G          1000#   1084
DIRBLK  000050R            1001    1002#
DIRECT  000000R     004    1001    1229#
DIRPTR  000044R            1001#
DLT    = 177762            1000#
DONE   = 000200            1000#   1071    1113
DRIVER  000560R            1001    1193#
DRT    = 177756            1000#
EOFSEN  000000R     002    1082*   1103*   1118*   1119    1193*   1214    1229#
EOMERR= ******  G          1000#
ERROR  = 100000            1000#   1071    1113
ETR    = 177766            1000#
FLNOTF= ******  G          1000#
FSTMOD= ******  G          1000#
HEADER  000074R     004    1229#
IFLMOD= ******  G          1000#
INVCMD= ******  G          1000#   1001    1002
ITOA   = ******  G         1000#   1229
LOOKUP  000040R     002    1001    1229#
LSTBLK  000046R            1001#
MES    = ******  G         1000#   1229
MTBA    000076R            1006#   1063*   1109*   1203*
MTCM    000072R            1004#   1033*   1070*   1071    1111*   1113    1206*
MTDO    000224R            1021    1027    1029    1060#
MTDO1   000260R            1071#   1072    1207
MTDO2   000272R            1034    1074#   1075    1083    1112
```

```
MTDRV   000164R          1017    1024    1030*   1104    1204
MTERR   000320R          1073    1080*
MTER1   000336R          1081    1084*   1128
MTST    000070R          1003*   1018    1069*   1074    1076    1080    1110*   1116    1126    1205*
MTWC    000074R          1005*   1060*   1061*   1106*   1201*
NXTBLK= ******  G        1000*
OPEN  = 000000           1000*   1229
PAK     000514R          1123    1135*   1212
PARAM   000000R          1001*   1012
PAREND  000050R          1001*
PC    =%000007           1000*   1013*   1017*   1022*   1023*   1024*   1034*   1035*   1055*   1062*   1078*   1079*   1104*
                         1108*   1112*   1123*   1125*   1136*   1143*   1144*   1202*   1204*   1207*   1212*   1220*   1229*
PIPFLG= ******  G        1000*   1216
PS    = 177776           1000*
PT11  = ****** U         1392
R8KCT   000042R          1001*
RDHDR   000342R          1002    1103*
RDONLY= 000000           1000    1001    1002    1011    1036    1064    1153    1196    1208    1229
READ    000036R          1001*
READBK= ******  G        1000*
RELCNT= ******  G        1000*   1054    1229
RESR7 = ******  G        1000*
REWIND  000106R          1002    1017*
RHDR1   000424R          1113*   1114
RHDR2   000436R          1116*
RHDR3   000502R          1115    1126*
RHDR4   000474R          1120    1124*
RHDR5   000346R          1104*   1121    1127
RHDR6   000464R          1117    1122*
RK11  = ****** U         1472
RP04  = ****** U         1610
RP11  = ****** U         1730
RST04 = ******  G        1000*   1143
RS04  = ****** U         1677
RX01  = ****** U         1802
R0    =%000000           1000*   1012*   1137*   1139    1140*
R1    =%000001           1000*   1138*   1139*
R2    =%000002           1000*   1229*
R3    =%000003           1000*   1020*   1025*   1028*   1030*   1031*   1032*   1033    1070    1229*
R4    =%000004           1000*   1135*
R5    =%000005           1000*   1026*   1031    1060    1111    1194    1206    1218*   1229*
R6    =%000006           1000*
SAV04 = ******  G        1000*   1135
SEARCH  000060R     002  1229*
SETMTO  000100R G        1010*
SHORT = ****** U         1229
SKIPR   000136R          1002    1024*
SP    =%000006           1000*   1122*   1229*
SRH   = 177770           1000*
TAB   = ******  G        1000*   1229
TA11  = ****** U         2008
TC11  = ****** U         1527
TM02  = ****** U         1231
TRBF    000000R     003  1053*   1054*   1063    1109    1137    1203    1229*
TRBSET  000206R          1053*   1062    1108    1136    1202
TRB_F   000002R     003  1053    1229*
```

```
TR79F = 000000            999
TXNAM = ******  G        1000*
UNIT    000022R          1001*
UNPACK= ******  G        1000*
UPACK1= ******  G        1000*
UPKNAM= ******  G        1000*   1229
WCOUNT  000026R          1001*
WRITE   000040R          1001*
WRTIDB  000156R          1022    1028*   1078
XBA   = 000004           1000*   1194
XBC   = 000016           1000*
XBKLGT= 177746           1000*   1229*
XBT   = 177754           1000*
XCM   = 000000           1000*
XCO   = 000010           1000*   1206
XDN   = 177776           1000*
XDR   = 000020           1000*   1229
XDT   = 000006           1000*
XFLCNT= 177722           1000*   1229*
XFLMOD= 177720           1000*   1229
XLSTBK= 177750           1000*
XNB   = 000022           1000*
XRD   = 000012           1000*   1031    1111
XSV   = 177774           1000*
XSVBLK= 177730           1000*
XSVCNT= 177726           1000*
XSVDAT= 177740           1000*   1229*
XSVEXT= 177736           1000*   1229*
XSVMAP= 177724           1000*
XSVNAM= 177732           1000*   1229*
XSVUPT= 177752           1000*
XSVXX = 177742           1000*
XWC   = 000002           1000*   1026*   1060    1218*
XWCTR = 177716           1000*
XWT   = 000014           1000*
XXNAM = 000024           1000*
XXXXX = ******  U           3
X1STBK= 177744           1000*
ZER   = 177760           1000*
$BUF  = ******  G        1000*   1107    1229
$BUF2 = ******  G        1000*
$TPNM1= ******  G        1000*   1229
$TXNAM= ******  G        1000*   1229
.     = 000126R      C04 1229*
```

DKCOMM	250#	
DKPARM	209#	
MTCOMM	820#	1229
MTPARM	225#	1002
PACK	54#	
PACK1	58#	
PARAM	170#	1001
RESREG	45#	1143
RESR5	48#	
RESR7	51#	
ROUTIN	32#	
SAVREG	42#	1135
TITLE	70#	1000
UNPACK	62#	
UPACK1	66#	

```
ADD     1054  1122  1213  1229
BEQ     1072  1075  1077  1081  1114  1117  1120  1229
BGT     1219
BIS     1020  1025
BIT     1018  1071  1074  1076  1080  1113  1116  1126
BMI     1073  1115
BNE     1019  1127  1142  1215  1217  1229
BPL     1229
BR      1021  1027  1029  1083  1121  1128  1229
CLR     1030  1069  1110  1205
CLRB    1103  1193
CMP     1119
INC     1140  1141  1229
INCB    1082  1118
JMP     1084  1229
JSR     1017  1022  1024  1034  1062  1078  1104  1108  1112  1123  1135  1136  1143  1202  1204
        1207  1212  1229
MOV     1012  1026  1028  1033  1053  1060  1063  1070  1105  1106  1107  1109  1111  1137  1138
        1194  1195  1201  1203  1206  1229
MOVB    1031  1139  1229
NEG     1061
RTS     1013  1023  1035  1055  1079  1125  1144  1220  1229
SUB     1218
SWAB    1032
TST     1124  1229
TSTB    1214  1216
.ASCIZ  1229
.BLKW   1229
.BYTE   1229
.CSECT  1229
.END    2308
.ENDC     31  1000  1001  1002  1050  1068  1177  1200  1211  1229  1230  1257  1321  1391  1471
        1482  1526  1537  1545  1609  1620  1676  1688  1729  1740  1801  1852  1898  1955  2007
        2053  2104  2135  2155  2211  2278  2287  2305  2307
.EVEN   1229
.GLOBL  1000
.IFDF      3   999  1001  1002  1231  1392  1472  1527  1610  1677  1730  1802  2008
.IFF    1001  1002
.IFNDF  1000  1036  1064  1153  1196  1208  1229  1253  1302  1478  1533  1542  1616  1684  1736
        1850  1879  1948  2047  2100  2129  2151  2205  2216  2284  2303
.IFT    1001  1002
.IIF    1000  1011  1229
.LIST      1
.MACRO    32    42    45    48    51    54    58    62    66    70   170   209   225   250   920
.NLIST     2
.PAGE   1000  1001  1008  1015  1221  1229
.SBTTL  1000  1001  1009  1016  1229
.TITLE  1000

. ABS.  000000  000
        000700  001
MTCOMM  000176  002
TRBF    002002  003
MTDIRT  000126  004
```

```
TR79F -  XXDP TR79F MODULE READ-ONLY    MACY11 27(732)  03-JAN-77  13:48  PAGE 4-1
DMQUFA.P12      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                       SEQ 0066


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *TR79FR,TR79FR.LST/CRF=DMQUFA.P12/EQ:TR79F/EQ:RDONLY
 RUN-TIME:  8  9  1 SECONDS
 RUN-TIME RATIO: 35/19=1.8
 CORE USED:  13K  (25 PAGES)


LNKX11    V022    3-JAN-77    13:51

*DMQUFA.BIN/T:20000,TRDP.MAP=TRDPM,TR79FR/E


LOAD MAP

TRANSFER ADDRESS: 000001
LOW LIMIT: 005760
HIGH LIMIT: 020000
**********
MODULE  TRDP
SECTION ENTRY   ADDRESS SIZE
(. ABS.)        000000  000000
(       )       005760  000710
        BCDCV   006240
        DATUPK  006356
        FSTMOD  006044
        $TPNM1  006162
<MTDIRT>        006670  000126
<RUNBUF>        007016  001000
<TRBF   >       010016  002002
<RESMON>        012020  004660
        BCLEAR  014134
        BKREAD  014004
        BMOVE   014124
        BUF     015420
        CHROUT  013016
        CLRBUF  014106
        CMPNAM  014164
        CRLF    012722
        DELAY   012222
        DEVERR  013236
        EOMERR  014050
        FLNOTF  014732
        IFLMOD  016422
        INVCMD  012644
        ITOA    013252
        MES     012662
        NXTBLK  013770
        PIPFLG  016543
        READBK  014024
        RELCNT  012672
        RESR5   014256
        RESR7   014042
        RESTRT  012070
        RST04   014072
```

```
        SAV04   014060
        TAB     012772
        TXNAM   016544
        UNPACK  014274
        UPACK1  014264
        UPKNAM  014232
        $BUF    014122
        $BUF2   014112
        $IDDB   013472
        $TXNAM  014176
**********
MODULE  TR79F
SECTION ENTRY   ADDRESS SIZE
        >               016700  000702
        SETMTO  017000
.MTCOMM>                017602  000176
        CRD4DR  017620
        CREWND  017604

 RUN-TIME:  1 SECONDS
```