TR79F

TRDP/XXDP TR79F MONITOR

MD-11-DMQUF-B

EP-DMQUF-B-DL-B

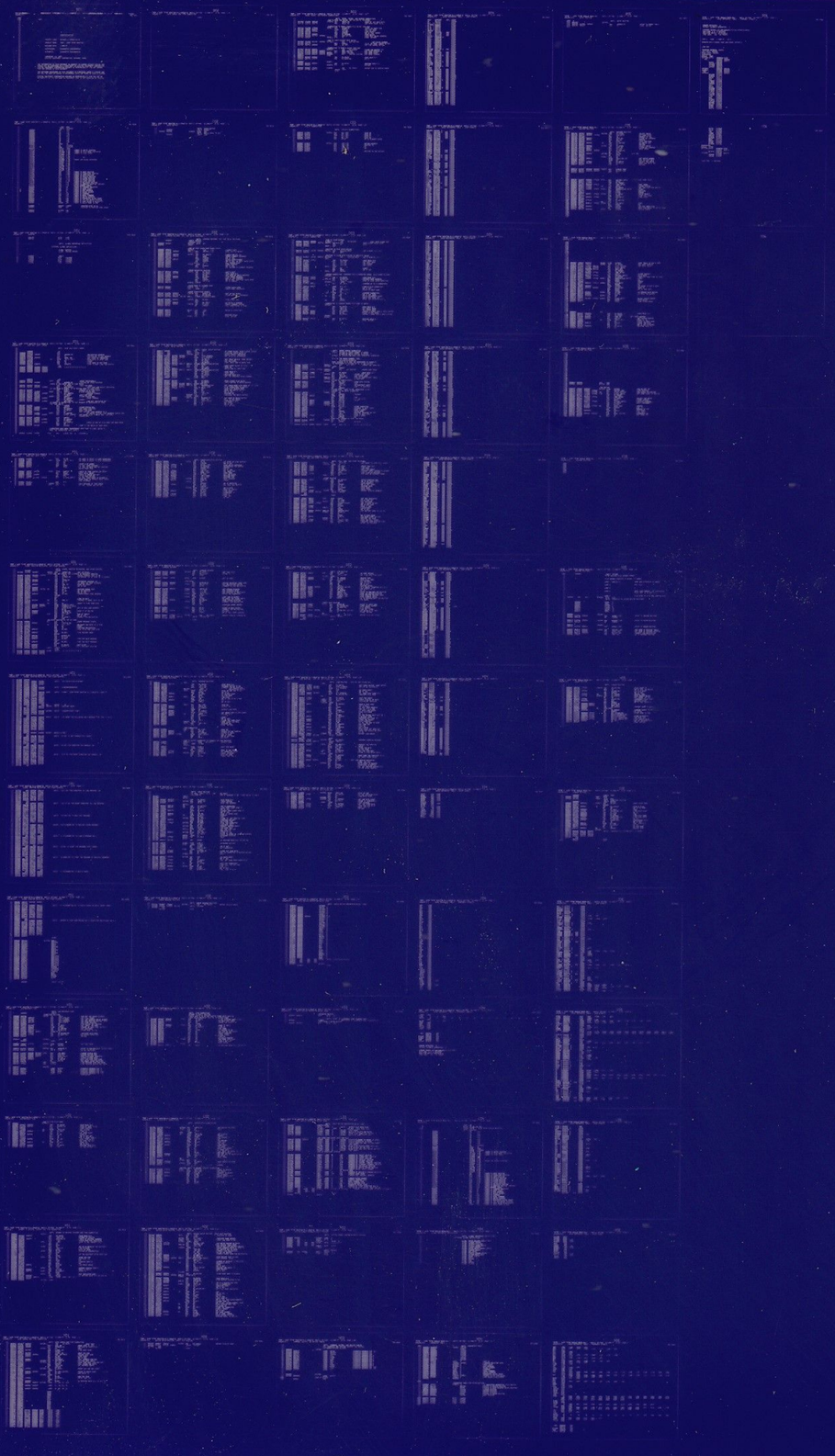
COPYRIGHT © 1977

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



```
  1252
  1333                 .REM    %
   (2)
   (2)
   (2)
   (2)
   (2)
   (2)
   (2)
   (2)
   (2)
   (2)                                      IDENTIFICATION
   (2)                                      --------------
   (2)
   (2)                      PRODUCT CODE:   MAINDEC-11-DMQUF-B-LA
   (2)
   (2)                      PRODUCT NAME:   TRDP - XXDP TR79F MONITOR
   (2)
   (2)                      RELEASE DATE:   1-MAR-77
   (2)
   (2)                      MAINTAINER:     DIAGNOSTIC ENGINEERING
   (2)
   (2)                      AUTHOR(S):      DIAGNOSTIC ENGINEERING
   (2)
   (2)
   (2)                      COPYRIGHT (C) 1977
   (2)                      DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.
   (2)
   (2)
   (2)
   (2)                 THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND
   (2)                 SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
   (2)                 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS
   (2)                 THAT MAY APPEAR IN THIS DOCUMENT.
   (2)
   (2)                 THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND
   (2)                 MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH A LICENSE.
   (2)
   (2)                 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR
   (2)                 RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
   (2)
   (2)                 %
```

```
(1)                                     .LIST   BIN
(1)                                     .SBTTL  EQUATES
(1)             000000C                 R0      =%0
(1)             000001                  R1      =%1
(1)             000002                  R2      =%2
(1)             000003                  R3      =%3
(1)             000004                  R4      =%4
(1)             000005                  R5      =%5
(1)             000006                  R6      =%6
(1)             000006                  SP      =%6
(1)             000007                  PC      =%7
(1)             000000                  OPEN    =0
(1)             177776                  PS      =177776
(1)             177776                  PSW     =177776
(1)             000340                  PRTY7   =340
(1)             177716                  XWCTR   =-62    ;INDEX TO WRITE COUNTER.
(1)             177720                  XFLMOD  =-60    ;INDEX TO FILE MODE INDICATOR
(1)             177722                  XFLCNT  =-56    ;INDEX TO FILE COUNT
(1)             177724                  XSVMAP  =-54    ;
(1)             177726                  XSVCNT  =-52    ;
(1)             177730                  XSVBLK  =-50    ;
(1)             177732                  XSVNAM  =-46    ;PHONY UFD BLOCK POINTERS
(1)             177736                  XSVEXT  =-42    ;
(1)             177740                  XSVDAT  =-40    ;
(1)             177742                  XSVXX   =-36    ;
(1)             177744                  X1STBK  =-34    ;
(1)             177746                  XBKLGT  =-32    ;
(1)             177750                  XLSTBK  =-30    ;
(1)             177752                  XSVUPT  =-26    ;
(1)             177754                  XBT     =-24    ;INDEX TO BOOT ROUTINE.
(1)             177756                  DRT     =-22    ;INDEX TO DIRECTORY ROUTINE
(1)             177760                  ZER     =-20    ;INDEX TO ZERO ROUTINE
(1)             177762                  DLT     =-16    ;INDEX TO DELETE ROUTINE
(1)             177764                  CLS     =-14    ;INDEX TO CLOSE ROUTINE
(1)             177766                  ETR     =-12    ;INDEX TO ENTER ROUTINE
(1)             177770                  SRH     =-10    ;INDEX TO LOKUP ROUTINE
(1)             177772                  ALC     =-6     ;INDEX TO ALLOCATE ROUTINE
(1)             177774                  XSV     =-4     ;INDEX TO SERVICE ROUTINE (DRIVER).
(1)             177776                  XDN     =-2     ;DRIVE NUMBER INDEX
(1)             000000                  XCM     =0      ;INDEX TO COMMAND REGISTER
(1)             000002                  XWC     =2      ;INDEX TO WORD COUNT
(1)             000004                  XBA     =4      ;INDEX TO BUS ADDRESS
(1)             000006                  XDT     =6      ;INDEX TO BLOCK NUMBER
(1)             000010                  XCO     =10     ;INDEX TO COMMAND
(1)             000012                  XRD     =12     ;INDEX TO READ COMMAND
(1)             000014                  XWT     =14     ;INDEX TO WRITE COMMAND
(1)             000016                  XBC     =16     ;INDEX TO REQUESTED BLOCK COUNT
(1)             000020                  XDR     =20     ;INDEX TO 1ST DIR BLOCK POINTER.
(1)             000022                  XNB     =22     ;INDEX TO LAST BLOCK # ALLOCATED.
(1)             000024                  XXNAM   =24     ;INDEX TO ASCII NAME IN DDB
(1)
(1)             010000                  MONCNT  =10000          ;MONITOR SIZE IS 4K
(1)             000024                  MNBK    =24             ;POINTER TO MONITOR CORE IMAGE.
(1)
(1)             000033                  ALTMOD  =33
(1)             000033                  ALT1    =33
```

```
(1)  000175                    ALT2    =175
(1)  000176                    ALT3    =176
(1)
(1)
(1)                            .SBTTL  GLOBAL REFERENCE DEFINITIONS
(1)
(1)                           ;EXTERNAL GLOBAL DEFINITIONS
(1)
(1)                            .GLOBL  SETMTO
(1)                            .GLOBL  CREWND,CRDHDR
(1)
(1)
(1)  000000                    .ASECT
(2)  164002                    MTST    =164002
(2)  164000                    MTCM    =164000
(2)  164004                    MTWC    =164004
(2)  164006                    MTBA    =164006
```

```
(2)                                             .SBTTL  ONCE ONLY BOOT LOADER
(2)
(2)             000000C                         .=0
(2)   000000    000240                          240
(2)   000002    012700  036000                  MOV     #36000,R0               ;RELOCATION ADDRESS
(2)   000006    005001                          CLR     R1                      ;INITIALIZE TO OLD ADDRESS
(2)   000010    012702  000400                  MOV     #256.,R2                ;SET COUNT OF TRANSFERS
(2)   000014    112120                  1$:     MOVB    (R1)+,(R0)+             ;MOVE BYTE TO UPER CORE
(2)   000016    005302                          DEC     R2                      ;COUNT
(2)   000020    001375                          BNE     1$                      ;LOOP BACK IF NOT DONE
(2)   000022    000137  036070                  JMP     @#36070                 ;JUMP TO BOOT LOAD SUB ROUTIN
(2)             000040                          .=40
(2)   000040    000000  000000  000000          .WORD   0,0,0,0,0,0,0,0,0,0,0,0
(2)   000046    000000  000000  000000
(2)   000054    000000  000000  000000
(2)   000062    000000  000000  000000
(2)
(2)
(2)
(2)
(2)             000070                          .=70
(2)   000070    000240                  INIT:   NOP                             ;RESET PERIPHERALS
(2)   000072    012706  040000                  MOV     #40000,SP               ;INITIALIZE STACK POINTER
(2)                                     ;CHECK IF MTU READY AND AT LOAD POINT
(2)   000076    004767  000160                  JSR     PC,READY                ;TEST FOR DONE
(2)   000102    032737  000040  164002          BIT     #40,@#MTST                      ;TEST FOR LOAD POINT INDICATOR
(2)                                     ;IF YES READ ID BURST, IF NO REWIND TAPE TO LOAD POINT
(2)   000110    001007                          BNE     BURST                   ;BRANCH IF @ LOAD POINT
(2)   000112    005037  164002          RWND:   CLR     @#MTST                  ;CLEAR STATUS REG.
(2)   000116    012737  000021  164000          MOV     #21,@#MTCM                      ;REWIND COMMAND
(2)   000124    004767  000132                  JSR     PC,READY                ;WAIT FOR DONE
(2)   000130    005037  164002          BURST:  CLR     @#MTST                  ;CLEAR STATUS REG.
(2)   000134    012737  000005  164000          MOV     #5,@#MTCM                       ;READ COMMAND
(2)   000142    004767  000114                  JSR     PC,READY                ;DONE??
(2)   000146    032737  000020  164002          BIT     #20,@#MTST                      ;IDB BIT INDICATOR SET ??
(2)                                     ;IO BURST IS TESTED FOR, IF NOT PRESENT AT BEGINNING OF TAPE
(2)                                     ;THE TAPE HALTS.
(2)   000154    001454                          BEQ     TAPERR
(2)   000156    005037  164002                  CLR     @#MTST                  ;CLEAR STATUS REG.
(2)   000162    012737  000005  164000          MOV     #5,@#MTCM               ;SET READ FUNCTION
(2)   000170    004767  000066                  JSR     PC,READY                ;WAIT FOR DONE
(2)   000174    005004                          CLR     R4                      ;SET INITIAL LOAD ADDRESS TO 0
(2)   000176    010437  164006          REED:   MOV     R4,@#MTBA                       ;R4 HOLDS INITIAL LOOP ADDRESS DURING DMA
(2)   000202    012737  177000  164004          MOV     #-512.,@#MTWC           ;SET BLOCK SIZE TO READ 256. (DECIMAL)
(2)                                             ;CHARACTERS
(2)   000210    005037  164002                  CLR     @#MTST
(2)   000214    012737  000005  164000          MOV     #5,@#MTCM
(2)   000222    004767  000034                  JSR     PC,READY
(2)   000226    032737  000010  164002          BIT     #10,@#MTST                      ;CHECK IF END OF FILE MARK HAS BEEN READ
(2)   000234    001402                          BEQ     3$
(2)   000236    000137  001000                  JMP     @#1000                  ;JUMP TO THE BEGIN OF TRDP PROG.
(2)
(2)                                     ;CHARACTERS HAVE BEEN TRANSFERRED FROM TAPE TO CORE, BEGIN TO
(2)                                     ;PACK THE CHARACTERS INTO WORDS.
(2)
(2)   000242    012703  177000          3$:     MOV     #-512.,R3               ;COUNTER
```

```
(2)  000246  010405                    MOV     R4,R5           ;R4 USED TO POINT TO WORD ADDRESSES
(2)  000250  112524            PACK:    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     READY:  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            TAPERR:  HALT                    ;HALT BECAUSE OF TAPE ERROR
(2)  000310  000207            RTN:     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  001434                        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$:     ASL     R1              ;READY TO TYPE CORE SIZE.
(1)  001102  116167  001236  000301        MOVB    KCODE(1),AK+1
(1)  001110  116167  001237  000274        MOVB    KCODE+1(1),AK+2
(1)  001116  166700  000112                SUB     LIMIT+2,R0      ;SET UP NEW LOAD ADDRESS
(1)  001122  010001                        MOV     R0,R1           ;
(1)  001124  010167  000652'               MOV     R1,RELCNT       ;SAVE IT AT RELTMP
(1)  001130  012702  002622                MOV     #$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  001415                        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  001457                        FHELP
(1)  001214  004567  000642'               JSR     R5,MES          ;TYPE THE HELP MESSAGE
(1)  001220  001546                        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-B 1-MAR-77"
(1)  001336  020055  054130  050104
(1)  001344  052040  033522  043071
(1)  001352  04644C  047117  052111
(1)  001360  051117  046440  030455
(1)  001366  026461  046504  052521
(1)  001374  026506  020102  026461
(1)  001402  040515  026522  033467
(1)  001410  020040  045440     000  AK:      .ASCIZ  '   K'
(1)  001415     045  042522  052123  ARSTRT:  .ASCIZ  '%A. START ADDR:'
(1)  001422  051101  020124  042101
(1)  001430  051104  000072
(1)  001434  044445  051516  043125  NOCORE:  .ASCIZ  '%INSUFFICIENT CORE'
(1)  001442  044506  044503  047105
(1)  001450  020124  047503  042522
(1)  001456     000
(1)  001457     045  047524  040440  FHELP:   .ASCIZ  '%TO ABORT THE FOLLOWING HELP MESSAGE TYPE CTRL C (↑C)%'
(1)  001464  047502  052122  052040
(1)  001472  042510  043040  046117
(1)  001500  047514  044527  043516
(1)  001506  044040  046105  020120
(1)  001514  042515  051523  043501
(1)  001522  020105  054524  042520
(1)  001530  041440  051124  020114
(1)  001536  020103  057050  024503
(1)  001544  000045
(1)  001546  022445  054524  042520  MNINST:  .ASCII'%%TYPE:'
(1)  001554     072
(1)  001555     045  036106  051103          .ASCII  '%F<CR> TO SET CONSOLE FILL COUNT'
(1)  001562  020076  047524  051440
(1)  001570  052105  041440  047117
(1)  001576  047523  042514  043040
(1)  001604  046111  020114  047503
(1)  001612  047125     124
(1)  001615     045  036104  051103          .ASCII  '%D<CR> FOR DIRECTORY ON CONSOLE, OR'
(1)  001622  020076  047506  020122
(1)  001630  044504  042522  052103
(1)  001636  051117  020131  047117
(1)  001644  041440  047117  047523
(1)  001652  042514  020054  051117
(1)  001660  042045  043057  041474          .ASCII  '%D/F<CR> FOR SHORT DIRECTORY ON CONSOLE, OR'
(1)  001666  037122  043040  051117
(1)  001674  051440  047510  052122
(1)  001702  042040  051111  041505
(1)  001710  047524  054522  047440
(1)  001716  020116  047503  051516
(1)  001724  046117  026105  047440
(1)  001732     122
```

```
(1)  001733     045  027504  036114          .ASCII  '%D/L<CR> FOR DIRECTORY ON LINE PRINTER, OR'
(1)  001740  051103  020076  047506
(1)  001746  020122  044504  042522
(1)  001754  052103  051117  020131
(1)  001762  047117  046040  047111
(1)  001770  020105  051120  047111
(1)  001776  042524  026122  047440
(1)  002004     122
(1)  002005     045  027504  027514          .ASCII  '%D/L/F<CR> FOR SHORT DIRECTORY ON LINE PRINTER,'
(1)  002012  036106  051103  020076
(1)  002020  047506  020122  044123
(1)  002026  051117  020124  044504
(1)  002034  042522  052103  051117
(1)  002042  020131  047117  046040
(1)  002050  047111  020105  051120
(1)  002056  047111  042524  026122
(1)  002064  051045  041440  050117          .ASCII  '%R COPY<CR> TO RUN COPY PROGRAM,'
(1)  002072  036131  051103  020076
(1)  002100  047524  051040  047125
(1)  002106  041440  050117  020131
(1)  002114  051120  043517  040522
(1)  002122  026115
(1)  002124  051045  043040  046111          .ASCII  '%R FILENAME<CR> TO RUN ANY OTHER PROGRAM.'
(1)  002132  047105  046501  036105
(1)  002140  051103  020076  047524
(1)  002146  051040  047125  040440
(1)  002154  054516  047440  044124
(1)  002162  051105  050040  047522
(1)  002170  051107  046501     056
(1)  002175     045  020114  044506          .ASCII  '%L FILENAME<CR> TO LOAD A PROGRAM ONLY'
(1)  002202  042514  040516  042515
(1)  002210  041474  037122  052040
(1)  002216  020117  047514  042101
(1)  002224  040440  050040  047522
(1)  002232  051107  046501  047440
(1)  002240  046116     131
(1)  002243     045  036123  051103          .ASCII  '%S<CR> TO START THE PROGRAM JUST LOADED,'
(1)  002250  020076  047524  051440
(1)  002256  040524  052122  052040
(1)  002264  042510  050040  047522
(1)  002272  051107  046501  045040
(1)  002300  051525  020124  047514
(1)  002306  042101  042105     054
(1)  002313     045  020123  042101          .ASCII  '%S ADDR<CR> TO START THE PROGRAM AT SPECIFIC ADDRESS'
(1)  002320  051104  041474  037122
(1)  002326  052040  020117  052123
(1)  002334  051101  020124  044124
(1)  002342  020105  051120  043517
(1)  002350  040522  020115  052101
(1)  002356  051440  042520  044503
(1)  002364  044506  020103  042101
(1)  002372  051104  051505     123
(1)  002377     045  020103  044506          .ASCII  '%C FILENAME<CR> TO RUN A CHAIN,'
(1)  002404  042514  040516  042515
(1)  002412  041474  037122  052040
(1)  002420  020117  052522  020116
```

```
(1)  002426  020101  044103  044501
(1)  002434  026116
(1)  002436  041445  043040  046111          .ASCII  '%C FILENAME/QV<CR> TO RUN A CHAIN IN QUICK VERIFY MODE.'
(1)  002444  047105  046501  027505
(1)  002452  053121  041474  037122
(1)  002460  052040  020117  052522
(1)  002466  020116  020101  044103
(1)  002474  044501  020116  047111
(1)  002502  050440  044525  045503
(1)  002510  053040  051105  043111
(1)  002516  020131  047515  042504
(1)  002524     056
(1)  002525     045  042522  042506          .ASCIZ  '%REFER TO XXDP USER MANUAL MD-11-DZQXA FOR ADDITIONAL HELP.%'
(1)  002532  020122  047524  054040
(1)  002540  042130  020120  051525
(1)  002546  051105  046440  047101
(1)  002554  040525  020114  042115
(1)  002562  030455  026461  055104
(1)  002570  054121  020101  047505
(1)  002576  020122  042101  044504
(1)  002604  044524  047117  046101
(1)  002612  044040  046105  027120
(1)  002620  000045
(1)                                         .EVEN
(1)
(1)  002622  000154'                $LITTB: $REL1
(1)  002624  001570'                        $REL2+2
(1)  002626  001550'                        $REL3
(1)  002630  000332'                        $REL4
(1)  002632  000334'                        $REL4+2
(1)  002634  000270'                        $REL5
(1)  002636  000272'                        $REL5+2
(1)  002640  000250'                        $REL6
(1)  002642  000252'                        $REL6+2
(1)  002644  001510'                        $REL7
(1)  002646  002540'                        $REL10+2
(1)  002650  000054'                        $REL11+2
(1)  002652  000374'                        $REL12+2
(1)  002654  000342'                        $REL13+2
(1)  002656  002276'                        $REL14+2
(1)  002660  002152'                        $REL15+2
(1)  002662  001462'                        $REL16+2
(1)  002664  000136'                        $COMC3
(1)  002666  000146'                        COMCON+2
(1)  002670  001032'                        GETIN0+2
(1)  002672  003120'                        LOAD4+2
(1)  002674  002102'                        $BUF
(1)  002676  002072'                        $BUF2
(1)  002700  002156'                        $TXNAM
(1)  002702  001452'                        $IDDB
(1)  002704  177777                         -1
(1)
(1)          000000'                        .CSECT  MTDIRT
```

```
(1)                                             .SBTTL  NON-RESIDENT CODE
(1)             000000'                         .CSECT
(1)                                     ;DIRECTORY ROUTINE.
(1)   000000' 005726                    NRDIR:  TST     (SP)+                   ;POP OFF 1ST STACK ELEMENT.
(1)   000002' 012667  002410'                   MOV     (SP)+,FILLCT            ;GET FILL COUNT.
(1)   000006' 012667  001466'                   MOV     (SP)+,CURDRV            ;GET THE CURRENT DRIVE NUMBER.
(1)   000012' 012667  000136'                   MOV     (SP)+,$COMC3            ;GET RES MONITOR RESTART ADDR.
(1)   000016' 012667  001124'                   MOV     (SP)+,KBPTR             ;GET KYBD POINTER.
(1)   000022' 005002                            CLR     R2                      ;FOR DEVICE SET ROUTINE.
(1)   000024' 004767  001442'                   JSR     PC,SETI                 ;SET INPUT. NO NAME.
(1)   000030' 105067  000030                    CLRB    FSTMOD                  ;ASSUME NO FAST MODE SWITCH.
(1)   000034' 004767  000034                    JSR     PC,NRGTSW               ;GET SWITCHES.
(1)   000040' 004567  002114'                   JSR     R5,BCLEAR               ;CLEAR NAME TO ?'S
(1)   000044' 004506' 000077  000011            IFNAM,77,9.
(1)   000052' 010565  177720                    MOV     R5,XFLMOD(R5)           ;INDICATE WILD MODE.
(1)   000056' 004775  177756                    JSR     PC,@DRT(R5)             ;GO OUTPUT DIRECTORY.
(1)   000062' 000207                            RTS     PC                      ;RETURN.
(1)
(1)   000064'    000            FSTMOD::        .BYTE   0
(1)             000066'                         .EVEN
(1)
(1)   000066' 105267  177772           SETFST: INCB    FSTMOD                  ;SET FAST MODE.
(1)   000072' 000400                            BR      NRGTSW
(1)
(1)   000074' 012701  000144'           NRGTSW: MOV     #NRSWTB,R1              ;POINT TO NON-RESIDENT SWITCH TABLE.
(1)   000100' 000167  000552'                   JMP     GTOKK                   ;GO SET SWITCHES.
(1)
(1)   000104' 016767  000030  001004'   LPSW:   MOV     LPS,MREG                ;CHANGE STATUS REG.
(1)   000112' 016767  000024  001000'           MOV     LPB,MOUT                ;CHANGE BUFFER REG.
(1)   000120' 052767  000200  002410'           BIS     #200,FILLCT             ;MAKE FILLCOUNT NEGATIVE.
(1)   000126' 012702  000014                    MOV     #14,R2                  ;OUTPUT A FORM FEED.
(1)   000132' 004767  000202'                   JSR     PC,DELAY                ;WAIT FOR FORM FEED DONE.
(1)   000136' 000756                            BR      NRGTSW                  ;SEE ABOUT OTHER SWITCHES.
(1)   000140' 177514                    LPS:    177514                          ;LINE PRINTER STAT REG ADDR.
(1)   000142' 177516                    LPB:    177516                          ;LINE PRINTER BUFFER REG ADDR.
(1)
(1)   000144'                           NRSWTB:
(2)   000144' 000066'                           .WORD   SETFST                  ;DISPATCH ADDRESS FOR /F
(2)   000154' 000104'                           .WORD   LPSW                    ;DISPATCH ADDRESS FOR /L
(2)   000164' 000212'                           .WORD   RTSPC                   ;DISPATCH ADDRESS FOR <15>
(2)   000172' 000546'                           .WORD   GTOK                    ;DISPATCH ADDRESS FOR <40>
(1)   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    @VMODE          ;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                 COMC01: 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,MES0         ;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 COMC03.
(1)  000136' 000140'                        $COMC3: COMC03
(1)
(1)                                         ;CHAIN MODE IS CLEARED HERE.
(1)  000140' 005027                         COMC03: 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         $REL1:  CLRBEG,0,CLREND-CLRBEG
(1)
(1)  000162' 004567  000454                 COMC02: JSR     R5,MES          ;TYPE A DOT 1ST.
(1)  000166' 004535'                                ADOT
(1)  000170' 004567  000630                         JSR     R5,INPUT        ;GO FETCH A COMMAND
(1)  000174' 004767  000346                 COMC05: JSR     PC,GTOK         ;CHECK COMMAND SYNTAX
(1)  000200' 000761                                 BR      COMCON          ;TILL EVERYTHING IS DONE
(1)
(1)
(1)  000202' 005046                         DELAY:: CLR     -(SP)           ;DELAY A LITTLE BIT.
(1)  000204' 005316                         1$:     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                 DO1:    INCB    BKCT            ;WANT FIRST BLOCK OF FILE.
(1)  000262' 010546                         DO2:    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'                                DO3:
(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     DO3             ;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.
```

F02


```
(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     COMC03          ;BACK TO COMC03.
(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  002612                     MOV     R4,RNBK         ;THAT TOO
(1)  000526' 000207                             RTS     PC              ;RETURN.
```

```
(1)
(1)                                             .SBTTL  COMMAND DECODER
(1)  000530' 004767  000654          SETCNT:    JSR     PC,ATOI             ;GET THE COUNT.
(1)  000534' 010267  003760                     MOV     R2,ICOUNT           ;STORE IT.
(1)  000540' 000402                             BR      GTSW
(1)
(1)  000542' 105267  177451          SETQV:     INCB    QVMODE              ;SET QV MODE.
(1)
(1)  000546'                         GTOK:GTSW:
(1)  000546' 012701  004542'                    MOV     #COMTAB,R1          ;DEVICE DECODING COMES HERE
(1)  000552' 016702  000074          GTOKK:     MOV     RELCNT,R2           ;GET RELOCATION FACTOR.
(1)  000556' 060201                             ADD     R2,R1               ;CORRECT R1 FOR RELOCATION.
(1)
(1)  000560' 016700  000340          GTOKX:     MOV     KBPTR,R0            ;GET STRING POINTER
(1)  000564' 012104                  GTOK1:     MOV     (1)+,R4             ;GET DISPATCH ADDR.
(1)  000566' 012103                             MOV     (1)+,R3             ;GET POINTER TO NEXT ENTRY.
(1)  000570' 122021                  GTOK2:     CMPB    (R0)+,(R1)+         ;MATCH THE CHARACTER?
(1)  000572' 001007                             BNE     GTOKY
(1)  000574' 105711                  GTOK3:     TSTB    (R1)                ;LAST CHR IN ENTRY?
(1)  000576' 001374                             BNE     GTOK2               ;NO,CHECK THE OTHER CHR'S
(1)  000600' 010067  000320                     MOV     R0,KBPTR            ;SAVE STRING POINTER.
(1)  000604' 060204                  GTOK4:     ADD     R2,R4               ;CORRECT FOR RELOCATION.
(1)  000606' 005002                             CLR     R2
(1)  000610' 000114                             JMP     (R4)                ;DISPATCH WHERE NEEDED.
(1)
(1)  000612' 010301                  GTOKY:     MOV     R3,R1               ;POINT TO NEXT ENTRY.
(1)  000614' 060201                             ADD     R2,R1               ;CORRECT FOR RELOCATION.
(1)  000616' 022711  177777                     CMP     #-1,(1)             ;FILLED?
(1)  000622' 001356                             BNE     GTOKX               ;BR IF YES. KEEP LOOKING.
(1)
(1)  000624' 004567  177260          INVCMD::           JSR     R5,COMCO1   ;REPORT INVALID COMMAND/SW
(1)  000630' 047111  041526  042115             .ASCIZ  'INVCMD/SW'
(1)  000638' 051457  000127
(1)                                             .EVEN
(1)
```

```
(1)                                             .SBTTL  MESSAGE ROUTINES
(2)  000642'                                    MES::
(2)  000642' 004467  001172                             JSR     R4,SAV04        ;SAVE REGS 0-4
(1)  000646' 012500                             MES0:   MOV     (R5)+,R0        ;MESSAGE BUFFER TO R0
(1)  000650' 062700                                     ADD     (PC)+,R0        ;CORRECT FOR RELOCATION.
(1)  000652' 000000                             RELCNT::       .WORD   0               ;RELOCATION FACTOR.
(1)  000654' 112002                                     MOVB    (R0)+,R2        ;PICK UP ONE CHR TO R2
(1)  000656' 001525                                     BEQ     GEX04           ;BR IF 0.
(1)  000660' 004767  000002                             JSR     PC,MES1         ;GO OUTPUT CHAR.
(1)  000664' 000773                                     BR      MES0            ;GO FOR MORE.
(1)  000666' 120227  000011                     MES1:   CMPB    R2,#11          ;TAB CODE?
(1)  000672' 001427                                     BEQ     TAB             ;BR IF YES TO DO A TAB.
(1)  000674' 120227  000045                             CMPB    R2,#'%          ;%?
(1)  000700' 001036                                     BNE     CHROUT          ;BR IF NOT.
(1)
(1)  000702' 010246                             CRLF::  MOV     R2,-(SP)        ;SAVE R2.
(1)  000704' 012702  005015                             MOV     #5015,R2        ;OUTPUT CR.
(1)  000710' 004767  000062                             JSR     PC,CHROUT
(1)  000714' 000302                                     SWAB    R2              ;OUTPUT LINE FEED.
(1)  000716' 004767  000054                             JSR     PC,CHROUT
(1)  000722' 116702  001462                             MOVB    FILLCT,R2       ;GET READY FOR FILLER CHARS.
(1)  000726' 005077  000046                     1$:     CLR     @MOUT           ;FILLER IS 0.
(1)  000732' 004767  000044                             JSR     PC,CROUT1
(1)  000736' 005302                                     DEC     R2              ;DONE?
(1)  000740' 003372                                     BGT     1$              ;BR IF NOT.
(1)  000742' 005027                                     CLR     (PC)+           ;CLEAR THE CHAR COUNT.
(1)  000744' 000000                             CHRCNT: .WORD   0               ;CHAR COUNT VARIABLE.
(1)  000746' 012602                             CRLF1:  MOV     (SP)+,R2        ;RESTORE R2.
(1)  000750' 000207                                     RTS     PC              ;DONE. RETURN.
(1)
(1)                                             ;TAB SUBROUTINE.
(1)  000752' 010246                             TAB::   MOV     R2,-(SP)        ;SAVE R2.
(1)  000754' 012702  000040                     1$:     MOV     #40,R2          ;SPACES DO THE TABBING.
(1)  000760' 004767  000012                             JSR     PC,CHROUT       ;OUTPUT A SPACE.
(1)  000764' 142767  000370  177752                     BICB    #370,CHRCNT     ;SEE IF DONE.
(1)  000772' 001370                                     BNE     1$              ;BR IF NOT DONE.
(1)  000774' 000764                                     BR      CRLF1           ;GO EXIT.
(1)
(1)                                             ;SUB TO OUTPUT CHARACTER TO CONSOLE OR LINE PRINTER
(1)  000776' 110237                             CHROUT::        MOVB    R2,@(PC)+       ;OUTPUT CHAR.
(1)  001000' 177566                             MOUT:   .WORD   177566
(1)  001002'                                    CROUT1:
(1)  001002' 105737                                     TSTB    @(PC)+          ;WAIT FOR READY.
(1)  001004' 177564                             MREG:   .WORD   177564
(1)  001006' 100375                                     BPL     CROUT1          ;BACK IF NOT READY.
(1)  001010' 105267  177730                             INCB    CHRCNT          ;UP CHARACTER COUNT.
(1)  001014' 004767  000116                     CROUT2: JSR     PC,CKYBD        ;CHECK KEYBOARD.
(1)  001020' 000207                                     RTS     PC              ;EXIT.NO CHAR.
(1)  001022' 000470                                     BR      GETCR1          ;CHECK FOR CTRL C.
```

```
(1)                                             .SBTTL  INPUT ROUTINE
(1)                                             ;TO CALL 'INPUT' DO A   JSR     R5,INPUT
(1)                                             ;FOLLOWED BY            +       ADR OF MESSAGE TO BE  TYPED PRIOR TO INPUT
(1)  001024'                            INPUT:
(2)  001024'                            GETIN:
(2)  001024'  004467  001010                    JSR     R4,SAV04                ;SAVE REGS 0-4
(1)  001030'  012700  003354'           GETIN0: MOV     #KBUF,R0                ;INPUT BUFFER
(1)  001034'  010001                            MOV     R0,R1                   ;SAVE THE ADDRESS
(1)  001036'                            GETI01:
(1)  001036'  004767  000120            2$:     JSR     PC,GETCHR               ;GET A CHARACTER
(1)  001042'  120227  000141            3$:     CMPB    R2,#141                 ;LESS THAN LOWER CASE A?
(1)  001046'  103405                            BLO     1$                      ;BR IF YES.
(1)  001050'  120227  000172                    CMPB    R2,#172                 ;HIGHER THAN LOWER CASE Z?
(1)  001054'  101002                            BHI     1$                      ;BR IF YES.
(1)  001056'  162702  000040                    SUB     #40,R2                  ;MAKE IT UPPER CASE.
(1)  001062'  120227  000177            1$:     CMPB    R2,#177                 ;RUBOUT?
(1)  001066'  001407                            BEQ     GETI02                  ;YES
(1)  001070'  110220                    GETI06: MOVB    R2,(R0)+                ;STORE IT. NOT SPECIAL CHAR.
(1)  001072'  120227  000015                    CMPB    R2,#15                  ;CARRIAGE RETURN?
(1)  001076'  001407                            BEQ     GETI08                  ;QUITTING TIME
(1)  001100'  004767  177562            GETI03: JSR     PC,MES1                 ;ECHO THE CHARACTER
(1)  001104'  000754                            BR      GETI01                  ;CONTINUE INPUT
(1)  001106'  020100                    GETI02: CMP     R1,R0                   ;RUBOUT, BUFFER EMPTY?
(1)  001110'  001752                            BEQ     GETI01                  ;YEP,ECHO CRLF
(1)  001112'  114002                    GETI11: MOVB    -(R0),R2                ;GET THE LAST CHR
(1)  001114'  000771                            BR      GETI03                  ;AND ECHO IT
(1)  001116'  112720  000012            GETI08: MOVB    #12,(R0)+               ;STORE LF TOO.
(1)  001122'  010127                            MOV     R1,(PC)+                ;POINT TO START OF KEYBOARD BUFFER.
(1)  001124'  000000                    KBPTR:  .WORD   0                       ;KEYBOARD POINTER.
(1)  001126'  004767  177550            GEX02:  JSR     PC,CRLF                 ;CRLF.
(2)  001132'                            GEX04:
(2)  001132'  000167  001100                    JMP     RESR5                   ;GO RESTORE REGS 0-4, DO RTS R5.
(1)
(1)  001136'  105737                    CKYBD:  TSTB    @(PC)+                  ;KEYBOARD ACTIVE?
(1)  001140'  177560                    KTKS:   .WORD   177560
(1)  001142'  100006                            BPL     CKYBD1                  ;BR IF NOT.
(1)  001144'  013702                            MOV     @(PC)+,R2               ;GET CHARACTER.
(1)  001146'  177562                    KTKB:   .WORD   177562
(1)  001150'  042702  177600                    BIC     #177600,R2              ;CLEAR OUT JUNK BITS.
(1)  001154'  062716  000002                    ADD     #2,(SP)                 ;SET UP CHARACTER IN BUFFER RETURN.
(1)  001160'  000207                    CKYBD1: RTS     PC                      ;EXIT.
(1)
(1)  001162'  005767  176754            GETCHR: TST     CHN                     ;IN CHAIN MODE?
(1)  001166'  001403                            BEQ     1$                      ;BR IF NOT.
(1)  001170'  004767  177220                    JSR     PC,CHAIN                ;YES. GET CHAR FROM CHAIN FILE.
(1)  001174'  000403                            BR      GETCR1
(1)  001176'  004767  177734            1$:     JSR     PC,CKYBD                ;WAIT FOR CHARACTER.
(1)  001202'  000775                            BR      1$                      ;LOOP TILL YOU GOT ONE.
(1)  001204'  120227  000003            GETCR1: CMPB    R2,#3                   ;CTRL C?
(1)  001210'  001363                            BNE     CKYBD1                  ;BR IF NOT.
(1)  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)
```

```
(1)                                             .SBTTL  GETNUM/ATOI SUBROUTINES
(1)
(1)
(1)  001306' 016700  177612          GETNUM: MOV     KBPTR,R0                ;GET STRING POINTER.
(1)  001312' 005001                  GTNM1:  CLR     R1                      ;DATA
(1)  001314' 005027                          CLR     (PC)+
(1)  001316' 000000                  YES:    .WORD   0
(1)  001320' 112002                  2$:     MOVB    (R0)+,R2                ;GET A BYTE
(1)  001322' 120227  000040                  CMPB    R2,#40                  ;SPACE?
(1)  001326' 001771                          BEQ     GTNM1                   ;YES,IGNORE IT
(1)  001330' 122702  000015                  CMPB    #15,R2                  ;CR?
(1)  001334' 001417                          BEQ     3$                      ;YES,RETURN
(1)  001336' 120227  000060                  CMPB    R2,#'0                  ;LOW LIMIT
(1)  001342' 002414                          BLT     3$                      ;TOO LOW
(1)  001344' 120227  000067                  CMPB    R2,#'7                  ;HIGH LIMIT
(1)  001350' 003011                          BGT     3$                      ;TOO HIGH
(1)  001352' 006301                          ASL     R1                      ;SHIFT OLD STUFF
(1)  001354' 006301                          ASL     R1                      ;3 TIMES LEFT
(1)  001356' 006301                          ASL     R1                      ;I.E. MULT. BY OCTAL 10
(1)  001360' 060201                          ADD     R2,R1                   ;ADD NEW TO OLD
(1)  001362' 162701  000060                  SUB     #'0,R1                  ;BUT GET RID OF ASCII STUFF
(1)  001366' 105267  177724                  INCB    YES                     ;SET FLAG
(1)  001372' 000752                          BR      2$                      ;MORE, MORE
(1)  001374'                         3$:
(1)  001374' 005300                  GTNM0:  DEC     R0
(1)  001376' 010067  177522                  MOV     R0,KBPTR                ;SAVE STRING POINTER.
(1)  001402' 105767  177710          GTNUM1: TSTB    YES
(1)  001406' 000207                          RTS     PC                      ;NOMORE
(1)
(1)                                  ;DECIMAL ASCII TO BINARY CONVERT SUBROUTINE.
(1)  001410' 016700  177510          ATOI:   MOV     KBPTR,R0                ;POINT TO STRING
(1)  001414' 112003                  ATOI1:  MOVB    (R0)+,R3                ;GET DIGIT.
(1)  001416' 162703  000060                  SUB     #'0,R3                  ;CONVERT TO BINARY
(1)  001422' 100764                          BMI     GTNM0                   ;NOT A DIGIT.
(1)  001424' 020327  000011                  CMP     R3,#9.                  ;CHECK UPPER LIMIT.
(1)  001430' 003361                          BGT     GTNM0                   ;TOO HIGH.
(1)  001432' 006302                          ASL     R2
(1)  001434' 006302                          ASL     R2
(1)  001436' 060302                          ADD     R3,R2                   ;ALL DONE.
(1)  001440' 000765                          BR      ATOI1
```

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

```
(1)  001672' 004567  176212          INVNAM: JSR     R5,COMCO1              ;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             READL1: 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)  002066' 004567  000022  CLRBUF::  JSR     R5,BCLEAR          ;CALL BYTE CLEAR SUB.
(1)  002072' 003402' 000000  000776  $BUF2:: BUF+2,0,$10.         ;DEST,CLEAR VALUE,COUNT
(1)  002100' 000207                   RTS     PC                 ;EXIT.
(1)  002102' 003400'                  $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.
(1)  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'                          UPKNM1:
(2)  002236'                          RESR5::
(2)  002236' 004767  177610                JSR     PC,RST04           ;RESTORE REGS 0-4
(1)  002242' 000205                        RTS     R5                 ;DONE. RETURN.
```

```
(1)                                                     .SBTTL  RAD50 UNPACK SUBROUTINE
(1)                                             ;       INPUT:  R0=ADR OF MOD40 NUMBER (2 WORDS)
(1)                                             ;               R1=ADR OF ASCII STRING (6 BYTES)
(1)                                             ;       OUTPUT: R1 POINTS ONE PAST LAST GENERATED CHARACTER
(1)                                             ;
(1)                                             ;       IF N IS THE MOD40 NUMBER, THEN
(1)                                             ;               N=C1*50↑2+C2*50+C3
(1)                                             ;       THUS, N/50↑2 IS C1 AND THE REMAINDER IS C2*50+C3
(1)                                             ;       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  000001     COEFF:  .WORD   1600.,40.,1.     ;40.↑2, 40.↑1,40.↑0
```

```
(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  176640             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,COMCO1               ;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)
```

```
(1)
(1)  002574' 004767 176642          DIR:    JSR     PC,SETI             ;NO NAME NEEDED.
(1)  002600' 012746                         MOV     (PC)+,-(SP)         ;GET FILE POSITION COUNT.
(1)  002602' 000001                 FILCNT: .WORD   1
(1)  002604' 004767 000000G                 JSR     PC,CREWND           ;REWIND TAPE.
(1)  002610' 004767 000000G         1$:     JSR     PC,CRDHDR           ;READ FILE LABEL.
(1)  002614' 000436                         BR      FLNOTF              ;EOT RETURN.
(1)  002616' 005316                         DEC     (SP)                ;GOT TO FILE?
(1)  002620' 001373                         BNE     1$                  ;BR IF NOT.
(1)  002622' 005726                         TST     (SP)+               ;RESTORE STACK.
(1)  002624' 012765 010000  000002          MOV     #MONCNT,XWC(R5)     ;4K'S WORTH. STARTING AT LOC 0
(1)  002632' 005065 000004                  CLR     XBA(R5)             ;XFR STARTS AT 0.
(1)  002636' 004767 177142                  JSR     PC,READBK           ;DO IT.
(1)  002642' 012701 000000'                 MOV     #NRDIR,R1           ;POINT TO NON-RES DIR ROUTINE.
(1)  002646' 016746 176252                  MOV     KBPTR,-(SP)         ;PASS THE BUFFER POINTER.
(1)  002652' 016746 175260                  MOV     $COMC3,-(SP)        ;PASS MONITOR RESTART ADDR.
(1)  002656' 016746 176604                  MOV     CURDRV,-(SP)        ;PASS CURRENT DRIVE.
(1)  002662' 000737                         BR      RUN40               ;GO TO NON-RESIDENT DIR ROUTINE
(1)                                                                     ;VIA RUN40.
(1)
(1)                                 ;INPUT INIT ROUTINE
(1)  002664' 016705 176562          INIT:   MOV     $IDDB,R5            ;POINT TO INPUT DDB.
(1)  002670' 010565 177744                  MOV     R5,X1STBK(R5)       ;DUMMY BLOCK NUMBER.
(1)  002674' 004775 177770                  JSR     PC,@SRH(R5)         ;FILE SEARCH.
(1)  002700' 000404                         BR      2$                  ;FILE NOT FOUND.
(1)  002702' 016565 177744  000006          MOV     X1STBK(R5),XDT(R5)  ;1ST BLOCK # TO INDT
(1)  002710' 000207                         RTS     PC                  ;FOUND. RETURN.
(1)  002712'                        2$:
(1)  002712' 004567 175172          FLNOTF::        JSR     R5,COMCO1   ;REPORT FILE NOT FOUND.
(1)  002716' 042516 043130  046111          .ASCIZ  'NEXFIL'
(1)  002724'    000
(1)          002726'                        .EVEN
(1)
```

```
(1)                                                 .SBTTL  LOAD ROUTINE. ;.BIN OR .BIC FILES ONLY
(2)  002726'                                    LOAD:
(2)  002726'  004767  176514                            JSR     PC,SETIN                ;SET INPUT DEVICE. NAME NEEDED.
(1)  002732'  012767  044502  001554                     MOV     #^BI,IFNAM+6            ;SET UP BIC EXTENSION.
(1)  002740'  112767  000103  001550                     MOVB    #'C,IFNAM+8.
(1)  002746'  005767  175170                            TST     CHN                     ;IN CHAIN MODE?
(1)  002752'  001005                                    BNE     1$                      ;BR IF YES.
(1)  002754'  112767  000077  001534                     MOVB    #'?,IFNAM+8.            ;NO. MAKE LAST CHAR WILD.
(1)  002762'  010565  177720                            MOV     R5,XFLMOD(R5)           ;INDICATE FILE MODE.
(2)  002766'                                    1$:
(2)  002766'  004767  177672                            JSR     PC,INITI                ;INIT FOR INPUT.
(1)  002772'  004767  176710                    LOAD1:  JSR     PC,GTDATA               ;INPUT A BLOCK OF DATA
(1)  002776'  005027                            LOAD2:  CLR     (PC)+                   ;INITIALIZE CHECKSUM
(1)  003000'  000000                            CHKSUM: .WORD   0
(1)  003002'  004767  000154                            JSR     PC,RDFRAM               ;READ A SYNC WORD
(1)  003006'  005703                                    TST     R3                      ;GOT A NULL?
(1)  003010'  001772                                    BEQ     LOAD2                   ;IF YES, KEEP READING.
(1)  003012'  005303                                    DEC     R3                      ;SEE IF IT'S A ONE
(1)  003014'  001025                                    BNE     CKSMER                  ;IF NOT, LOAD ERROR.
(1)  003016'  004767  000140                            JSR     PC,RDFRAM               ;SYNC IS A WORD
(1)  003022'  105703                                    TSTB    R3                      ;OF 1
(1)  003024'  001021                                    BNE     CKSMER                  ;SO THE SECOND HALF MUST BE 0
(1)  003026'  004767  000154                            JSR     PC,RD2FRM               ;2 BYTES=1 WORD
(1)  003032'  010304                                    MOV     R3,R4                   ;ASSUMING NOT DONE YET
(1)  003034'  162704  000004                            SUB     #4,R4                   ;MINUS THE HEADER
(1)  003040'  022704  000002                            CMP     #2,R4                   ;BYTE COUNT=6?
(1)  003044'  001431                                    BEQ     LJMP                    ;IT IS, THE END IS NEAR
(1)  003046'  004767  000134                            JSR     PC,RD2FRM               ;GET LOAD ADR
(1)  003052'  010302                                    MOV     R3,R2                   ;INTO R2
(1)  003054'  004767  000102                    LOAD3:  JSR     PC,RDFRAM               ;GET A BYTE
(1)  003060'  100016                                    BPL     LOAD4                   ;BYTE COUNT NOT ZERO YET
(1)  003062'  105767  177712                            TSTB    CHKSUM                  ;CHECK SUM SHOULD BE ZERO
(1)  003066'  001743                                    BEQ     LOAD2                   ;IT IS
(1)  003070'  004567  175014                    CKSMER: JSR     R5,COMCO1               ;REPORT LOAD ERROR.
(1)  003074'  045503  046523  051105                     .ASCIZ  'CKSMER'
(1)  003102'     000
(1)           003104'                                   .EVEN
(1)  003104'  004567  175000                    POFLOW: JSR     R5,COMCO1               ;PROGRAM OVERFLOW MESSAGE.
(1)  003110'  047520  046106  000117                     .ASCIZ  'POFLO'
(1)                                                     .EVEN
(1)  003116'  020227  000000'                    LOAD4:  CMP     R2,#R6STCK              ;PROTECT THE MONITOR
(1)  003122'  103370                                    BHIS    POFLOW                  ;ABORT
(1)  003124'  110322                                    MOVB    R3,(R2)+                ;STORE THE BYTE.
(1)  003126'  000752                                    BR      LOAD3                   ;GO GET MORE
(1)  003130'  004767  000052                    LJMP:   JSR     PC,RD2FRM               ;GET THE JUMP ADR
(1)  003134'  010327                                    MOV     R3,(PC)+                ;STORE IT FOR RAINY DAYS
(1)  003136'  000001                            STADR:  .WORD   1
(1)  003140'  004767  000016                            JSR     PC,RDFRAM               ;MAKE SURE THE CHECKSUM IS OK
(1)  003144'  105767  177630                            TSTB    CHKSUM                  ;WE CHECK EVERY BLOCK
(1)  003150'  001347                                    BNE     CKSMER
(1)  003152'  112737  000004  000041                     MOVB    #4,@#41         ;SET LOAD MEDIUM INDICATOR.
(1)  003160'  000207                                    RTS     PC                      ;DONE. GET OUT.
(1)  003162'  005301                            RDFRAM: DEC     R1                      ;BYTE COUNT IN BUFFER
(1)  003164'  100003                                    BPL     RDFRMA                  ;SOMETHING IN BUFFER
(1)  003166'  004767  176514                            JSR     PC,GTDATA               ;NO, GET ANOTHER BUFFER FULL
(1)  003172'  000773                                    BR      RDFRAM                  ;DO THE HOUSE KEEPING
```

```
(1)  003174' 112003                RDFRMA: MOVB    (R0)+,R3            ;PICK UP CHR
(1)  003176' 060367  177576                ADD     R3,CHKSUM           ;DO THE CHECKSUM STUFF
(1)  003202' 005304                        DEC     R4                  ;LOAD BYTE COUNT
(1)  003204' 000207                RDFRMB: RTS     PC
(1)  003206' 004767  177750        RD2FRM: JSR     PC,RDFRAM           ;GET ONE BYTE FIRST
(1)  003212' 010327                        MOV     R3,(PC)+            ;STORE IT TEMPORARILY
(1)  003214' 000000                LTEMP:  .WORD   0                   ;TEMP STORAGE.
(1)  003216' 004767  177740                JSR     PC,RDFRAM           ;GET THE OTHER BYTE
(1)  003222' 110367  177767                MOVB    R3,LTEMP+1          ;INTO THE HIGH BYTE
(1)  003226' 016703  177762                MOV     LTEMP,R3            ;BACK INTO R3
(1)  003232' 000207                        RTS     PC                  ;RETURN
```

```
(1)                                         .SBTTL  BATCH DEVICE DESCRIPTOR BLOCK (DDB)
(1)  003234'                            BTCDDB:
(1)  003234'  000000                    RWCTR:  0
(1)  003236'  000000                    RFLMOD: 0
(1)  003240'  000000                    RFLCNT: 0
(1)  003242'  000000                    RSVMAP: 0
(1)  003244'  000000                    RSVCNT: 0
(1)  003246'  000000                    RSVBLK: 0
(1)  003250'  000000  000000            RSVNAM: 0,0
(1)  003254'  000000                    RSVEXT: 0
(1)  003256'  000000                    RSVDAT: 0
(1)  003260'  000000                    RSVXX:  0
(1)  003262'  000000                    R1STBK: 0
(1)  003264'  000000                    RBKLGT: 0
(1)  003266'  000000                    RLSTBK: 0
(1)  003270'  000000                    RSVUPT: 0
(1)  003272'  000000                    RBOOT:  0
(1)  003274'  000000                    RDRT:   0
(1)  003276'  000000                    RZER:   0
(1)  003300'  000000                    RDLT:   0
(1)  003302'  000000                    RCLS:   0
(1)  003304'  000000                    RRETR:  0
(1)  003306'  000000                    RSRCH:  0
(1)  003310'  000000                    RALC:   0
(1)  003312'  000000                    RSRV:   0
(1)  003314'  000000                    RDRV:   0
(1)  003316'  000000                    RCM:    0
(1)  003320'  000000                    RWC:    0
(1)  003322'  000000                    RBA:    0
(1)  003324'  000000                    RDT:    0
(1)  003326'  000000                    RCOM:   0
(1)  003330'  000000                    RPRC:   0
(1)  003332'  000000                    RRPWC:  0
(1)  003334'  000000                    RRBKCT: 0
(1)  003336'  000000                    RDIR:   0
(1)  003340'  000000                    RNBK:   0
(1)  003342'     000     000     000    RFNAM:  .BYTE   0,0,0,0,0,0,0,0,0
(1)  003345'     000     000     000
(1)  003350'     000     000     000
(1)  003353'                            BTCEND:
(1)           003354'                           .EVEN
```

```
(1)
(1)  003354' 000012                  ;KEYBOARD BUFFER.
(1)                                  KBUF:   .BLKW   10.
(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.
```

```
(1)                                     .SBTTL  INPUT DEVICE DESCRIPTOR BLOCK (DDB)
(1)  004400'                            DDBSTR:INDDB:
(1)  004400' 000000                     IWCTR:  .WORD   0       ;XWCTR    OUTPUT FILE OPEN FLAG
(1)  004402' 000000                     IFLMOD::        .WORD   0       ;XFLMOD   FILE MODE FLAG
(1)  004404' 000000                     IFLCNT: .WORD   0       ;XFLCNT   FILE COUNT
(1)  004406' 000000                     ISVMAP: .WORD   0       ;XSVMAP   BLK # OF FILE'S MAP BLOCK
(1)  004410' 000000                     ISVCNT: .WORD   0       ;XSVCNT   ENTRY # OF CURR FILE IN UFD
(1)  004412' 000000                     ISVBLK: .WORD   0       ;XSVBLK   BLK # OF CURR FILE'S UFD
(1)
(1)                                     ;UFD DIRECTORY ENTRY DATA FOR FILE
(1)  004414' 000000  000000             ISVNAM: .WORD   0,0     ;XSVNAM   FILE'S FILENAME IN RAD50 (2 WORDS)
(1)  004420' 000000                     ISVEXT: .WORD   0       ;XSVEXT   FILE'S EXTENSION IN RAD50
(1)  004422' 000000                     ISVDAT: .WORD   0       ;XSVDAT   FILE'S CREATION DATE IN DOS FORMAT
(1)  004424' 000000                     ISVXX:  .WORD   0       ;XSVXX    (NOT USED?)
(1)  004426' 000000                     IISTBK: .WORD   0       ;XISTBK   BLOCK # OF FILE'S FIRST DATA BLK
(1)  004430' 000000                     IBKLGT: .WORD   0       ;XBKLGT   # OF BLOCKS IN THE FILE
(1)  004432' 000000                     ILSTBK: .WORD   0       ;XLSTBK   BLOCK # OF LAST DATA BLOCK WRITTEN
(1)  004434' 000000                     ISVUPT: .WORD   0       ;XSVUPT   (NOT USED?)
(1)                                     ;END OF DIRECTORY ENTRIES
(1)
(1)  004436' 000000                     INBOOT: .WORD   0       ;XBT      ADDRESS OF "BOOT" ROUTINE
(1)  004440' 000000                     INDRT:  .WORD   0       ;DRT      ADDRESS OF "DIRECTORY" ROUTINE
(1)  004442' 000000                     INZER:  .WORD   0       ;ZER      ADDRESS OF "ZERO" ROUTINE
(1)  004444' 000000                     INDLT:  .WORD   0       ;DLT      ADDRESS OF "DELETE" ROUTINE
(1)  004446' 000000                     INCLS:  .WORD   0       ;CLS      ADDRESS OF "CLOSE" ROUTINE
(1)  004450' 000000                     INETR:  .WORD   0       ;ETR      ADDRESS OF "ENTER" (CREATE) ROUTINE
(1)  004452' 000000                     INSRH:  .WORD   0       ;SRH      ADDRESS OF "LOOKUP" (SEARCH) ROUTINE
(1)  004454' 000000                     INALC:  .WORD   0       ;ALC      ADDRESS OF "ALLOCATE" ROUTINE
(1)  004456' 000000                     INSRV:  .WORD   0       ;XSV      ADDRESS OF DEVICE DRIVER ROUTINE
(1)  004460' 000000                     INDRV:  .WORD   0       ;XDN      CURRENT DRIVE (UNIT) NUMBER
(1)  004462'                            INDEV:                  ;R5 POINTS HERE
(1)  004462' 000000                     INCM:   .WORD   0       ;XCM      ADDRESS OF DEVICE'S COMMAND REGISTER
(1)  004464' 000000                     INWC:   .WORD   0       ;XWC      CURRENT WORD COUNT
(1)  004466' 000000                     INBA:   .WORD   0       ;XBA      CURRENT BUS (MEMORY) ADDRESS
(1)  004470'                            INBLK:
(1)  004470' 000000                     INDT:   .WORD   0       ;XDT      CURRENT BLOCK NUMBER
(1)  004472' 000000                     INCOM:  .WORD   0       ;XCO      CURRENT COMMAND CODE
(1)  004474' 000000                     INPRC:  .WORD   0       ;XRD      READ COMMAND CODE
(1)  004476' 000000                     INPWC:  .WORD   0       ;XWT      WRITE COMMAND CODE
(1)  004500' 000000                     INBKCT: .WORD   0       ;XBC      REQUESTED BLOCK COUNT
(1)  004502' 000000                     INDIR:  .WORD   0       ;XDR      ADDRESS OF FIRST DIRECTORY BLK #
(1)  004504' 000000                     INNBK:  .WORD   0       ;XNB      LAST BLOCK # ALLOCATED (NEXT BLK #)
(1)  004506'    000     000     000     IFNAM:  .BYTE   0,0,0,0,0,0,0,0,0       ;XXNAM    FILE'S NAME IN ASCII (9 CHAR'S)
(1)  004511'    000     000     000
(1)  004514'    000     000     000
(1)          004520'                            .EVEN
(1)  004520'                            DDBEND:
```

```
                                        .SBTTL  INITIALIZABLE VARIABLES/ASCII STRINGS
                                
004520' 000000                  ICOUNT: .WORD   0
004522'    000                  RUNID:  .BYTE   0
004523'    000                  PIPFLG::        .BYTE   0
004524'    000     000     000  TXNAM:: .BYTE   0.0,0.0,0,0
004527'    000     000     000
004532'    000     000     000  TXEXT:  .BYTE   0,0,0
004535'                         CLREND:

                                        ;ASCII STRINGS
004535'    045  000056          ADOT:   .ASCIZ  '%.'
004540'    011     000          ATAB:   .BYTE   11,0
                                        .EVEN
```

```
(1)                                             .SBTTL  COMMAND, SWITCH, AND DEVICE TABLES
(1)                                             ;ALL COMMANDS ARE CHECKED AGAINST THE QUOTES
(1)                                             ;NO ABBREVIATIONS ALLOWED
(1)  004542'                                COMTAB:
(2)  004542' 000542'                            .WORD   SETQV           ;DISPATCH ADDRESS FOR /QV
(2)  004552' 000530'                            .WORD   SETCNT          ;DISPATCH ADDRESS FOR /
(2)  004560' 000144'                            .WORD   COMCON          ;DISPATCH ADDRESS FOR .
(2)  004566' 002406'                            .WORD   FILL            ;DISPATCH ADDRESS FOR F
(2)  004574' 002442'                            .WORD   START           ;DISPATCH ADDRESS FOR S
(2)  004602' 002726'                            .WORD   LOAD            ;DISPATCH ADDRESS FOR L
(2)  004610' 002500'                            .WORD   RUN             ;DISPATCH ADDRESS FOR R
(2)  004616' 000214'                            .WORD   DOIT            ;DISPATCH ADDRESS FOR C
(2)  004624' 002574'                            .WORD   DIR             ;DISPATCH ADDRESS FOR D
(2)  004632' 000212'                            .WORD   RTSPC           ;DISPATCH ADDRESS FOR E
(2)  004640' 000212'                            .WORD   RTSPC           ;DISPATCH ADDRESS FOR <15>
(2)  004646' 000546'                            .WORD   GTOK            ;DISPATCH ADDRESS FOR <40>
(1)  004654' 177777                             .WORD   -1
(1)
(1)
(1)  004656'                                DEVTAB:
(1)  004656' 000000G                            .WORD   SETMTO
1334                                        ;END OF SOURCE.
1335            000001                          .END
```

```
ROOT    004535R     005  1333#
RK      001410           1333#*
RLC   = 177772           1333#
RLTMOD= 000033           1333#
RLT1  = 000033           1333#
RLT2  = 000175           1333#
RLT3  = 000176           1333#
RRSTRT  001415           1333#
RTAB    004540R     005  1333#
RTOI    001410R     005  1333#
RTOI1   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#
CHAIND  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#*
DOBEND  004520R     005  1333#
DOBSTR  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   001457           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#
GETI11  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    000546P     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#
INDOB   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#
ISVX)   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#
MES1    000666R      005  1333#
MNBK  = 000024            1333#
MNINST  001546            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  001434            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      00*  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#
RFLMOD  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#
SAV04   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#
SRH    = 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    = 177775           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#
XSVDAT= 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#
X1STBK= 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#
$COMC3  000136R     005   1333#*
$DEC    000702R           1333#
$FEB    000606R           1333#
$IDDB   001452R G   005   1333#
$JAN    000600R           1333#
$JUL    000644R           1333#
$JUN    000636R           1333#
$LITTB  002622            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
RESRS	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	109#	1333
$BOOT	1254#	1333
$IDENT	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
```

K04

```
TRDP - XXDP TR79F MONITOR M-11-DMQUF-B  MACY11 27(732)  01-MAR-77  10:34  PAGE 4-1
DMQUFB.P11      CROSS REFERENCE TABLE -- PERMANENT SYMBOLS                    SEQ 0049

.LIMIT   1333
.LIST       3        4    1333
.MACRO      5       14      23      26      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.  002706     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=DMQUFB.P11/EQ:TRDP
 RUN-TIME:  3 4 .6 SECONDS
 RUN-TIME RATIO: 295/9=32.5
 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,CRLF
(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
```

```
                        .SBTTL  PARAMETER TABLE
000000' 000000G  PARAM:  INVCMD
000002' 000000'          DIRECT
000004' 000000G          INVCMD
000006' 000000G          INVCMD
000010' 000000G          INVCMD
000012' 000000G          INVCMD
000014' 000040'          LOOKUP
000016' 000000G          INVCMD
000020' 000570'          DRIVER
000022' 000000   UNIT:    0                    ;UNIT #
000024' 164000   CMDREG:  164000               ;COMMAND REGISTER ADDR
000026' 000000   WCOUNT:  0                    ;WORD COUNT
000030' 000000   BUSADR:  0                    ;BUS ADDRESS
000032' 000000   BLOCK:   0                    ;BLOCK NUMBER
000034' 000000   COMD:    0                    ;COMMAND
000036' 000005   READ:    5                    ;READ COMMAND
000040' 000003   WRITE:   3                    ;WRITE COMMAND
000042' 000000   RBKCT:   0                    ;REQUESTED BLOCK COUNT
000044' 000050'  DIRPTR:  DIRBLK               ;POINTS TO 1ST DIR BLOCK.
000046' 000000   LSTBLK:  0                    ;LAST BLOCK # ALLOCATED
000050'          PAREND:


                        ;PARAMETER TABLE FOR MTCOM SECTION.
000050' 177777   DIRBLK: -1                    ;-1 INDICATES SEQUENTIAL FILE DEVICE.
000052' 000106'          REWIND                ;POINTS TO REWIND ROUTINE.
000054' 000142'          SKIPR                 ;POINTS TO SKIP REVERSE BLOCK ROUTINE.
000056' 000000G          INVCMD                ;CLRHDR POINTER.
000060' 000000G          INVCMD                ;WRTHDR POINTER.
000062' 000000G          INVCMD                ;WRTEOF POINTER.
000064' 000352'          RDHDR                 ;POINTS TO RDHDR ROUTINE.
000066' 000000G          INVCMD                ;WEOT POINTER

000070' 164002   MTST:    164002
000072' 164000   MTCM:    164000
000074' 164004   MTWC:    164004
000076' 164006   MTBA:    164006
```

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

```
                                                .SBTTL  MAGTAPE ROUTINES
000106' 004767  000062          REWIND: JSR     PC,MTDRV                ;SELECT DRIVE
000112' 032777  000040  177750          BIT     #40,@MTST               ;AT LOAD PIONT
000120' 001005                          BNE     1$                      ;BRANCH IF YES
000122' 042703  000377                  BIC     #377,R3                 ;CLEAR OUT GARBAGE
000126' 052703  000021                  BIS     #21,R3                  ;REWIND
000132' 000440                          BR      MTDO                    ;CHCK FOR DONE
000134' 004767  000026          1$:     JSR     PC,WRTIDB               ;GO WRITE IDB
000140' 000207                          RTS     PC                      ;RETURN
000142' 004767  000026          SKIPR:  JSR     PC,MTDRV                ;SELECT DRIVE
000146' 042703  000377                  BIC     #377,R3                 ;CLEAR OUT GARBAGE
000152' 052703  000011                  BIS     #11,R3                  ;BACK SPACE
000156' 012765  000001  000002          MOV     #1,XWC(R5)
000164' 000423                          BR      MTDO
000166' 012703  000005          WRTIDB: MOV     #5,R3           ;W/R IDB
000172' 000420                          BR      MTDO            ;
000174' 005003                  MTDRV:  CLR     R3
000176' 116503  000013                  MOVB    XRD+1(R5),R3            ;GET DRIVE NUMBER
000202' 000303                          SWAB    R3                      ;PUT IN LEFT HALF
000204' 010377  177662                  MOV     R3,@MTCM                ;SELECT THE DRIVE.
000210' 004767  000066                  JSR     PC,MTDO2                ;WAIT FOR DRIVE RDY
000214' 000207                          RTS     PC                      ;RETURN.


000216' 012767  000002' 000000' TRBSET: MOV     #TRBUF,TRBF             ;
000224' 066767  000000G 000000'         ADD     RELCNT,TRBF                     ;
000232' 000207                          RTS     PC                      ;




000234' 016577  000002  177632  MTDO:   MOV     XWC(R5),@MTWC           ;GET WORD COUNT
000242' 005477  177626                  NEG     @MTWC
000246' 004767  177744                  JSR     PC,TRBSET               ;
000252' 016777  000000' 177616          MOV     TRBF,@MTBA              ;MEM. ADDR.
000260' 005077  177604          1$:     CLR     @MTST                   ;CLEAR STATUS
000264' 010377  177602                  MOV     R3,@MTCM                ;SET THE FUNCTION AND GO
000270' 032777  100200  177574  MTDO1:  BIT     #ERROR+DONE,@MTCM       ;DONE?
000276' 001774                          BEQ     MTDO1                   ;WAITING
000300' 100413                          BMI     MTERR                   ;ERROR
000302' 032777  000001  177560  MTDO2:  BIT     #1,@MTST                ;DRIVE READY?
000310' 001774                          BEQ     MTDO2                   ;LOOP TILL DONE
000312' 032777  000040  177550          BIT     #40,@MTST               ;LOAD PIONT
000320' 001402                          BEQ     1$                      ;
000322' 004767  177640                  JSR     PC,WRTIDB               ;
000326' 000207                  1$:     RTS     PC
000330' 032777  000010  177532  MTERR:  BIT     #10,@MTST               ;EOF?
000336' 001403                          BEQ     MTER1                   ;NO
000340' 105267  000000'                 INCB    EOFSEN                  ;SET EOF SEEN FLAG
000344' 000756                          BR      MTDO2                   ;WAIT FOR DRIVE TO SETTLE DOWN
000346' 000167  000000G         MTER1:  JMP     DEVERR                  ;DEVICE ERROR.
```

```
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
000352' 105067 000000'  RDHDR:  CLRB    EOFSEN          ;KEEP TRACK OF EOF'S SEEN
000356' 004767 177612   RHDR5:  JSR     PC,MTDRV        ;SEE WHAT HAPPENS
000362' 012767 177762 000176    MOV     #-14.,BYTCNT    ;SET BYTE COUNT
000370' 012777 177762 177476    MOV     #-14.,@MTWC
000376' 016767 000000G 000160   MOV     $BUF,DES
000404' 004767 177606           JSR     PC,TRBSET       ;
000410' 016777 000000' 177460   MOV     TRBF,@MTBA
000416' 005077 177446           CLR     @MTST           ;CLEAR INHIBIT
000422' 016577 000012 177442    MOV     XRD(R5),@MTCM   ;DO A READ
000430' 004767 177646           JSR     PC,MTD02        ;TEMP FIX MAKE SURE OF DONE
000434' 032777 100200 177430    BIT     #ERROR+DONE,@MTCM ;DONE?
000442' 001774                  BEQ     RHDR1           ;NO
000444' 100422                  BMI     RHDR3           ;ERROR
000446' 032777 000010 177414    BIT     #10,@MTST       ;EOF?
000454' 001407                  BEQ     RHDR6           ;NO
000456' 105267 000000'          INCB    EOFSEN          ;UP COUNT OF EOF'S SEEN.
000462' 022767 000003 000000'   CMP     #3,EOFSEN       ;SEEN 3?
000470' 001405                  BEQ     RHDR4           ;BR IF YES. EOT.
000472' 000731                  BR      RHDR5
000474' 062716 000002   RHDR6:  ADD     #2,@SP          ;NO ERRORS SKIP RETURN
000500' 004767 000020           JSR     PC,PAK          ;GO PACK THE BUFFER
000504' 005767 000000G  RHDR4:  TST     BUF             ;
000510' 000207                  RTS     PC
000512' 032777 001000 177350    BIT     #1000,@MTST     ;BLOCK LENGTH ERROR?
000520' 001316                  BNE     RHDR5           ;DATA BLOCK
000522' 000711                  BR      MTER1           ;SOME KINDA ERROR.
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
000524'                 PAK:
(1) 000524' 004467 000000G      JSR     R4,SAV04        ;SAVE REGS 0-4
1138 000530' 004767 177462      JSR     PC,TRBSET       ;
1139 000534' 016700 000000'     MOV     TRBF,R0         ;GET INPUT BUFFER
1140 000540' 016701 000020      MOV     DES,R1          ;GET OUTPUT BUFFER
1141 000544' 112021     1$:     MOVB    (R0)+,(R1)+     ;PACK IT IN
1142 000546' 005200             INC     R0              ;POINT TO NEXT WORD
1143 000550' 005267 000012      INC     BYTCNT          ;COUNT THE BYTE
1144 000554' 001373             BNE     1$              ;BR IF NOT FINISHED
1145 000556' 004767 000000G     JSR     PC,RST04        ;RESTORE REGS 0-4
1146 000562' 000207             RTS     PC
```

```
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 1191  000564' 000000                          DES:     000
 1192  000566' 000000                          BYTCNT:  000
 1193
 1194                                          ;THE DRIVER
 1195  000570' 105067  000000'                 DRIVER:  CLRB     EOFSEN                  ;SAW NO EOF YET
 1196  000574' 016567  000004  177762                   MOV      XBA(R5),DES             ;LOAD POINTER
 1197  000602' 012767  177000  177756  1$:              MOV      #-512.,BYTCNT           ;256 WORDS PER RECORD
 1203  000610' 012777  177000  177256  11$:             MOV      #-512.,@MTWC            ;SET WORD COUNT IN DEVICE
 1204  000616' 004767  177374                           JSR      PC,TRBSET               ;
 1205  000622' 016777  000000' 177246                   MOV      TRBF,@MTBA
 1206  000630' 004767  177340                           JSR      PC,MTDRV                ;SELECT THE DRIVE.
 1207  000634' 005077  177230                           CLR      @MTST                   ;CLEAR INHIBIT
 1208  000640' 016577  000010  177224                   MOV      XCO(R5),@MTCM           ;EXECUTE COMMAND
 1209  000646' 004767  177416                           JSR      PC,MTDO1                ;WAIT FOR DONE
 1214  000652' 004767  177646                           JSR      PC,PAK
 1215  000656' 062767  001000  177700  2$:              ADD      #512.,DES               ;UPDATE
 1216  000664' 105767  000000'                          TSTB     EOFSEN                  ;EOF ???
 1217  000670' 001007                                   BNE      3$                      ;BRANCH IF YES
 1218  000672' 105767  000000G                          TSTB     PIPFLG                  ;PIP MODE?
 1219  000676' 001004                                   BNE      3$                      ;YES, EXIT
 1220  000700' 162765  000400  000002                   SUB      #256.,XWC(R5)           ;DECREMENT WC
 1221  000706' 003335                                   BGT      1$                      ;NOT DONE YET
 1222  000710' 000207                          3$:      RTS      PC
```

```
1224
1225
1226
1227
1228
1229
1230
```

```
1231            000000'                    .CSECT  MTCOMM
 (1)                             ;         TR79F MAGTAPE FILE HEADER FORMAT
 (1)
 (1)                             ;         FILE HEADER CONSISTS OF SEVEN (7) WORDS.
 (1)
 (1)                             ;         WORD 1 THROUGH 3              NAME AND EXTENSION IN RAD50
 (1)
 (1)                             ;         WORD 4                        PPN (401)
 (1)
 (1)                             ;         WORD 5                        0
 (1)
 (1)                             ;         WORD 6                        DATE (DOS FORMAT). IF CONTIGUOUS
 (1)                             ;                                       FILE, BIT 15 IS SET TO 1.
 (1)
 (1)                             ;         WORD 7                        FILE SIZE IN # OF 256 WORD BLOCKS.
 (1)   000000'    000      000   EOFSEN:   .BYTE   0,0
 (1)                                       .SBTTL  DISPATCH TO UNIQUE DEVICE ROUTINES.
 (1)
 (1)            000000                     CDIR=   0
 (1)            000002                     CRWD=   2
 (1)            000004                     CSKPR=  4
 (1)            000006                     CCLHD=  6
 (1)            000010                     CWTHD=  10
 (1)            000012                     CWEOF=  12
 (1)            000014                     CRHD=   14
 (1)
 (1)   000002'                   CREWND::
 (1)   000002' 012746  000002              MOV     #CRWD,-(SP)           ;POINT TO REWIND ROUTINE.
 (1)   000006' 000405                      BR      COMMON
 (1)
 (1)   000010' 012746  000004    CSKIPR:   MOV     #CSKPR,-(SP)          ;POINT TO SKIPR ROUTINE.
 (1)   000014' 000402                      BR      COMMON
 (1)
 (1)   000016'                   CRDHDR::
 (1)   000016' 012746  000014              MOV     #CRHD,-(SP)           ;POINT TO RDHDR ROUTINE.
 (1)
 (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  000000G             ADD     RELCNT,(SP)           ;CORRECT FOR RELOCATION.
 (1)   000036' 012607                      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
2310            000001                          .END
```

```
ALC    = 177772            1000#
BCDCV  = ******  G          1000#
BCLEAR= ******  G          1000#
BKREAD= ******  G          1000#
BLOCK   000032R            1001#
BMOVE  = ******  G          1000#
BUF    = ******  G          1000#   1126    1231
BUSADR  000030R            1001#
BYTCNT  000566R            1107*   1143*   1192#   1197*
CCLHD  = 000006            1231#
CDIR   = 000000            1231#
CHROUT= ******  G          1000#   1231
CLRBUF= ******  G          1000#
CLS    = 177764            1000#
CMDREG  000024R            1001#
CMPNAM= ******  G          1000#   1231
COMD    000034R            1001#
COMMON  000022R     002    1231#
CRDHDR  000016R G   002    1231#
CREWND  000002R G   002    1231#
CRHD   = 000014            1231#
CRLF   = ******  G          1000#   1231
CRWD   = 000002            1231#
CSKIPR  000010R     002    1231#
CSKPR  = 000004            1231#
CWEOF  = 000012            1231#
CWTHD  = 000010            1231#
DATUPK= ******  G          1000#   1231
DELAY  = ******  G          1000#
DES     000564R            1109*   1140    1191#   1196*   1215*
DEVERR= ******  G          1000#   1086
DIRBLK  000050R            1001    1002#
DIRECT  000000R     004    1001    1231#
DIRPTR  000044R            1001#
DLT    = 177762            1000#
DONE   = 000200            1000#   1073    1115
DRIVER  000570R            1001    1195#
DRT    = 177756            1000#
EOFSEN  000000R     002    1084*   1105*   1120*   1121    1195*   1216    1231#
EOMERR= ******  G          1000#
ERROR  = 100000            1000#   1073    1115
ETR    = 177766            1000#
FLNOTF= ******  G          1000#
FSTMOD= ******  G          1000#
HEADER  000074R     004    1231#
IFLMOD= ******  G          1000#
INVCMD= ******  G          1000#   1001    1002
ITOA   = ******  G          1000#   1231
LOOKUP  000040R     002    1001    1231#
LSTBLK  000046R            1001#
MES    = ******  G          1000#   1231
MTBA    000076R            1006#   1065*   1111*   1205*
MTCM    000072R            1004#   1035*   1072*   1073    1113*   1115    1208*
MTDO    000234R            1022    1029    1031    1062#
MTDO1   000270R            1073#   1074    1209
MTDO2   000302R            1036    1076#   1077    1085    1114
```

```
MTDRV   000174R          1017    1025    1032*   1106    1206
MTERR   000330R          1075    1082*
MTER1   000346R          1083    1086*   1130
MTST    000070R          1003*   1018    1071*   1076    1078    1082    1112*   1118    1128    1207*
MTWC    000074R          1005*   1062*   1063*   1108*   1203*
NXTBLK= ******  G        1000*
OPEN  = 000000           1000*   1231
PAK     000524R          1125    1137*   1214
PARAM   000000R          1001*   1012
PAREND  000050R          1001*
PC    =%000007           1000*   1013*   1017*   1023*   1024*   1025*   1036*   1037*   1057*   1064*   1080*   1081*   1106*
                         1110*   1114*   1125*   1127*   1138*   1145*   1146*   1204*   1206*   1209*   1214*   1222*   1231*
PIPFLG= ******  G        1000*   1218
PS    = 177776           1000*
PT11  = ****** U         1394
RBKCT   000042R          1001*
RDHDR   000352R          1002    1105*
RDONLY= 000000           1000    1001    1002    1011    1038    1066    1155    1198    1210    1231
READ    000036R          1001*
READBK= ******  G        1000*
RELCNT= ******  G        1000*   1056    1231
RESR7 = ******  G        1000*
REWIND  000106R          1002    1017*
RHDR1   000434R          1115*   1116
RHDR2   000446R          1118*
RHDR3   000512R          1117    1128*
RHDR4   000504R          1122    1126*
RHDR5   000356R          1106*   1123    1129
RHDR6   000474R          1119    1124*
RK11  = ****** U         1474
RP04  = ****** U         1612
RP11  = ****** U         1732
RST04 = ******  G        1000*   1145
RS04  = ****** U         1679
RX01  = ****** U         1804
R0    =%000000           1000*   1012*   1139*   1141    1142*
R1    =%000001           1000*   1140*   1141*
R2    =%000002           1000*   1231*
R3    =%000003           1000*   1020*   1021*   1026*   1027*   1030*   1032*   1033*   1034*   1035    1072    1231*
R4    =%000004           1000*   1137*
R5    =%000005           1000*   1028*   1033    1062    1113    1196    1208    1220*   1231*
R6    =%000006           1000*
SAV04 = ******  G        1000*   1137
SEARCH  000060R     002  1231*
SETMTO  000100R G        1010*
SHORT = ****** U         1231
SKIPR   000142R          1002    1025*
SP    =%000006           1000*   1124*   1231*
SRH   = 177770           1000*
TAB   = ******  G        1000*   1231
TA11  = ****** U         2010
TC11  = ****** U         1529
TM02  = ****** U         1233
TRBF    000000R     003  1055*   1056*   1065    1111    1139    1205    1231*
TRBSET  000216R          1055*   1064    1110    1138    1204
TRBUF   000002R     003  1055    1231*
```

```
TR79F = 000000            999
TXNAM = ******  G         1000#
UNIT    000022R           1001#
UNPACK= ******  G         1000#
UPACK1= ******  G         1000#
UPKNAM= ******  G         1000#   1231
WCOUNT  000026R           1001#
WRITE   000040R           1001#
WRTIDB  000166R           1023    1030#   1080
XBA   = 000004            1000#   1196
XBC   = 000016            1000#
XBKLGT= 177746            1000#   1231*
XBT   = 177754            1000#
XCM   = 000000            1000#
XCO   = 000010            1000#   1208
XDN   = 177776            1000#
XDR   = 000020            1000#   1231
XDT   = 000006            1000#
XFLCNT= 177722            1000#   1231*
XFLMOD= 177720            1000#   1231
XLSTBK= 177750            1000#
XNB   = 000022            1000#
XRD   = 000012            1000#   1033    1113
XSV   = 177774            1000#
XSVBLK= 177730            1000#
XSVCNT= 177726            1000#
XSVDAT= 177740            1000#   1231*
XSVEXT= 177736            1000#   1231*
XSVMAP= 177724            1000#
XSVNAM= 177732            1000#   1231*
XSVUPT= 177752            1000#
XSVXX = 177742            1000#
XWC   = 000002            1000#   1028*   1062    1220*
XWCTR = 177716            1000#
XWT   = 000014            1000#
XXNAM = 000024            1000#
XXXXX = ****** U             3
X1STBK= 177744            1000#
ZER   = 177760            1000#
$BUF  = ******  G         1000#   1109    1231
$BUF2 = ******  G         1000#
$TPNM1= ******  G         1000#   1231
$TXNAM= ******  G         1000#   1231
.     = 000126R     004   1231#
```

DKCOMM	250#	
DKPARM	209#	
MTCOMM	820#	1231
MTPARM	225#	1002
PACK	54#	
PACK1	58#	
PARAM	170#	1001
RESREG	45#	1145
RESR5	48#	
RESR7	51#	
ROUTIN	32#	
SAVREG	42#	1137
TITLE	70#	1000
UNPACK	62#	
UPACK1	66#	

```
ADD     1056    1124    1215    1231
BEQ     1074    1077    1079    1083    1116    1119    1122    1231
BGT     1221
BIC     1020    1026
BIS     1021    1027
BIT     1018    1073    1076    1078    1082    1115    1118    1128
BMI     1075    1117
BNE     1019    1129    1144    1217    1219    1231
BPL     1231
BR      1022    1029    1031    1085    1123    1130    1231
CLR     1032    1071    1112    1207
CLRB    1105    1195
CMP     1121
INC     1142    1143    1231
INCB    1084    1120
JMP     1086    1231
JSR     1017    1023    1025    1036    1064    1080    1106    1110    1114    1125    1137    1138    1145    1204    1206
        1209    1214    1231
MOV     1012    1028    1030    1035    1055    1062    1065    1072    1107    1108    1109    1111    1113    1139    1140
        1196    1197    1203    1205    1208    1231
MOVB    1033    1141    1231
NEG     1063
RTS     1013    1024    1037    1057    1081    1127    1146    1222    1231
SUB     1220
SWAB    1034
TST     1126    1231
TSTB    1216    1218
.ASCIZ  1231
.BLKW   1231
.BYTE   1231
.CSECT  1231
.END    2310
.ENDC     31    1000    1001    1002    1052    1070    1179    1202    1213    1231    1232    1259    1323    1393    1473
        1484    1528    1539    1547    1611    1622    1678    1690    1731    1742    1803    1854    1890    1957    2009
        2055    2106    2137    2157    2213    2280    2289    2307    2309
.EVEN   1231
.GLOBL  1000
.IFDF      3     999    1001    1002    1233    1394    1474    1529    1612    1679    1732    1804    2010
.IFF    1001    1002
.IFNDF  1000    1038    1066    1155    1198    1210    1231    1255    1304    1480    1535    1544    1618    1686    1738
        1852    1881    1950    2049    2102    2131    2153    2207    2218    2286    2305
.IFT    1001    1002
.IIF    1000    1011    1231
.LIST      1
.MACRO    32      42      45      48      51      54      58      62      66      70     170     209     225     250     820
.NLIST     2
.PAGE   1000    1001    1008    1015    1223    1231
.SBTTL  1000    1001    1009    1016    1231
.TITLE  1000

. ABS.  000000    000
        000712    001
MTCOMM  000176    002
TRBF    002002    003
MTDIRT  000126    004
```

```
 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *TR79FR,TR79FR.LST/CRF=DMQUFB.P12/EQ:TR79F/EQ:RDONLY
 RUN-TIME:  2  2  .3 SECONDS
 RUN-TIME RATIO:  167/5=28.6
 CORE USED:  13K  (25 PAGES)


LNKX11   V022   1-MAR-77   10:41

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


LOAD MAP

TRANSFER ADDRESS: 000001
LOW LIMIT: 005750
HIGH LIMIT: 020000
**********
MODULE  TRDP
SECTION ENTRY   ADDRESS SIZE
(. ABS.)        000000  000000
(       )       005750  00071C
        BCDCV   006230
        DATUPK  006346
        FSTMOD  006034
        $TPNM1  006152
(MTDIRT)        006660  000126
(RUNBUF)        007006  001000
(TRBF   )       010006  002002
(RESMON)        012010  004660
        BCLEAR  014124
        BKREAD  013774
        BMOVE   014114
        BUF     015410
        CHROUT  013006
        CLRBUF  014076
        CMPNAM  014154
        CRLF    012712
        DELAY   012212
        DEVERR  013226
        EOMERR  014040
        FLNOTF  014722
        IFLMOD  016412
        INVCMD  012634
        ITOA    013242
        MES     012652
        NXTBLK  013760
        PIPFLG  016533
        READBK  014014
        RELCNT  012662
        RESR5   014246
        RESR7   014032
        RESTRT  012060
```

```
        RSTO4    014062
        SAVO4    014050
        TAB      012762
        TXNAM    016534
        UNPACK   014264
        UPACK1   014254
        UPKNAM   014222
        SBUF     014112
        SBUF2    014102
        SIDDB    013462
        STXNAM   014166
**********
MODULE  TR79F
SECTION ENTRY    ADDRESS SIZE
(       )        016670  000712
        SETMTO   016770
(MTCOMM)         017602  000176
        CRDHDR   017620
        CREWND   017604

 RUN-TIME:  0 SECONDS
```