MPG

COMMON SECTIONS (MPG)

MD-11-DTECA-B

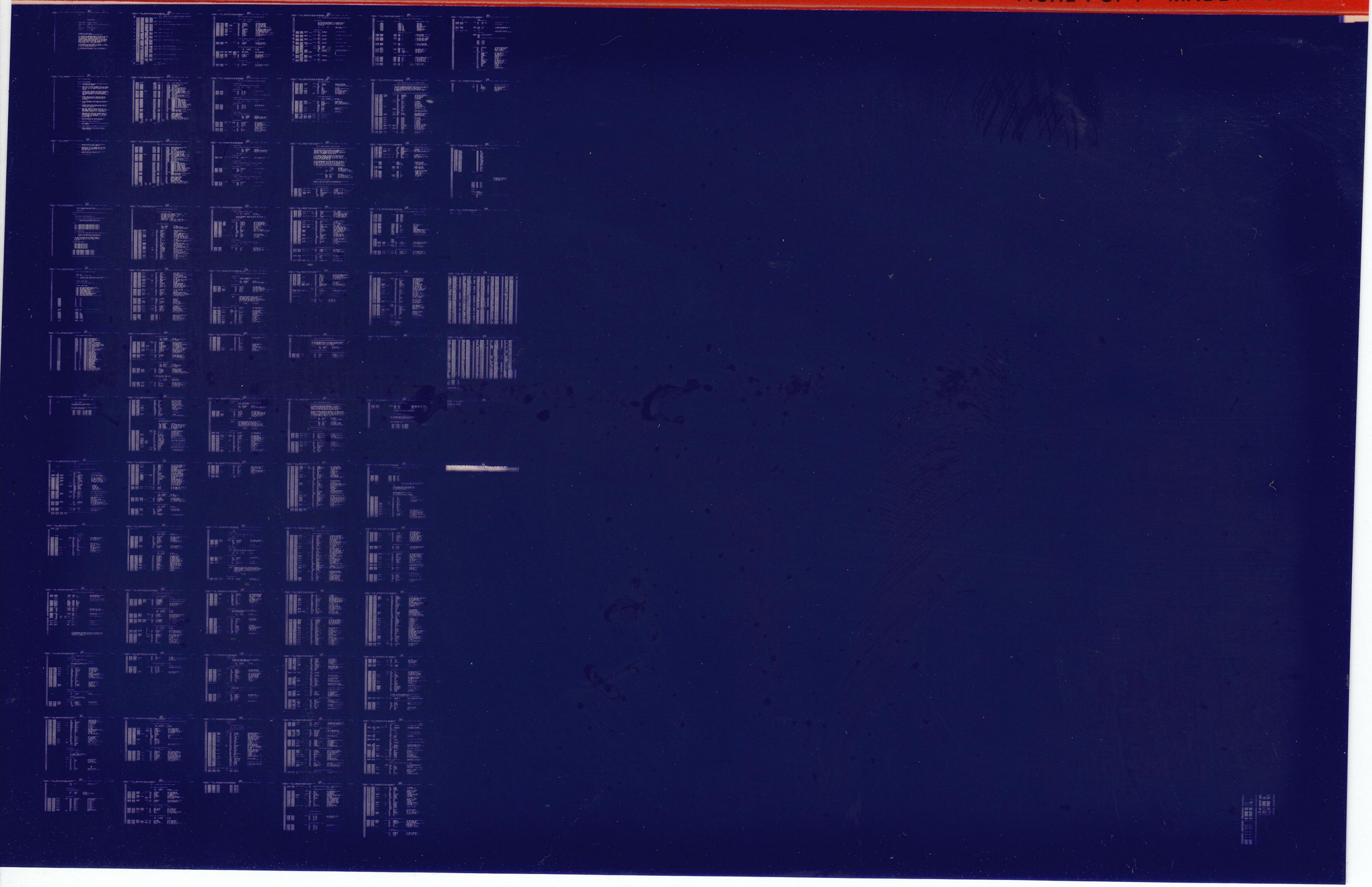
EP-DTECA-B-DL-B

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.



```
                .REM    %

                        IDENTIFICATION
                        --------------

PRODUCT CODE:   MAINDEC-11-DTECA-B
PRODUCT NAME:   COMMON SECTIONS FOR ALL MPG EXECUTIVES
DATE:           APRIL 1976
MAINTAINER:     SYSTEMS RELIABILITY
AUTHOR:         C. E. HARPER
```



```
                %
```

```
                        .SBTTL  REVISION HISTORY


                ;
                ;       APR 76  DTECA-B RELEASE
                ;
                ;       JAN 76  ELIMINATED THE PRELIMINARY FILE SEARCH FUNCTION
                ;               FOR THE DELETE COMMAND.
                ;
                ;       NOV 75  ADDED TEST FOR MEMORY MANAGEMENT AND THE MESSAGE
                ;               THAT ASKS IF IT IS TO BE USED. ALSO MADE CHANGES
                ;               THAT WILL LOAD THE CORRECT VERSION OF THE MPG
                ;               PROGRAM.
                ;
                ;       NOV 75  CHANGED CONSOLE TERMINAL AND PRINTER HANDLERS SO
                ;               THAT THEY GO TO MPG WHEN THEY NEED TO ACCESS THE
                ;               USER PROGRAM FLAGWORD. ALSO ADDED NECESSARY
                ;               INTERFACE ADDRESSES IN THE MPG INTERFACE ADDRESS
                ;               TABLE.
                ;
                ;       NOV 75  REMOVED THE CODE FOR THE USER READ FUNCTION
                ;               IN THE CONSOLE TERMINAL CONTROL ROUTINE AND
                ;               INTERRUPT SERVICING. THIS FUNCTION WAS NOT
                ;               BEING USED.
                ;
                ;       NOV 75  IN "↑C" PROCESSING CODE AND ONE TIME HOUSEKEEPING,
                ;               CHANGED PSW ACCESS TO BE COMPATIBLE WITH THE
                ;               LSI-11.
                ;
                ;       NOV 75  UPDATED INTERNAL AND EXTERNAL GLOBAL NAMES FOR
                ;               COMPATIBILITY WITH THE NEW DIRECTORY ROUTINES
                ;               IN THE DEVICE HANDLERS.
                ;
                ;       NOV 75  MADE MISC. CHANGES TO REDUCE CORE SIZE. FLAG
                ;               WORDS WERE MOVED IN-LINE WITH THEIR CODE. FLAG BITS
                ;               WERE CHANGED TO MAKE USE OF "TST/TSTB/BMI/BPL"
                ;               INSTRUCTIONS INSTEAD OF "BIT" INSTRUCTIONS.
                ;
                ;       NOV 75  ADDED CODE THAT STORES THE FILENAME FOUND WHEN
                ;               SEARCHING FOR A FILE WHOSE SUPPLIED NAME MAY
                ;               HAVE CONTAINED WILD CHARACTERS (?). THE ADDRESS
                ;               OF THE STORAGE AREA WAS ADDED TO MPG'S EXEC
                ;               INTERFACE TABLE.
                ;
                ;       NOV 75  MOVED HOUSEKEEPING OF VECTOR LOCATIONS 300
                ;               THRU 776 TO MPG'S ONE TIME HOUSEKEEPING.
                ;
                ;       NOV 75  REMOVED ALL "RESET" INSTRUCTIONS.
                ;
                ;       NOV 75  ADDED COMMENTS CONCERNING THE NEW RP04/RP05/RP06 AND
                ;               RX11 HANDLERS.
                ;
                ;       OCT 75  CONSOLE TERMINAL NOW ECHOES "$" FOR AN ALTMODE.
                ;
                ;       OCT 75  REMOVED HARDWARE TAB CODE FROM THE CONSOLE
                ;               TERMINAL ROUTINES. WILL NOW ALWAYS SIMULATE
                ;               A TAB WITH SPACES.
```

```
                                ;
                                ;       OCT 75  REMOVED TYPING OF A CR/LF WHEN ENCOUNTERING
                                ;               A 00 BYTE ON THE CONSOLE TERMINAL.
                                ;
                                ;       OCT 75  REMOVED 'DEV NOT RDY' MESSAGE FROM THE PAPER
                                ;               TAPE HANDLER. WILL NOW REPORT 'DEV ERR'. ALSO
                                ;               ADDED CODE THAT IGNORES LEADING 0'S ON THE
                                ;               FIRST DATA BLOCK.
                                ;
                                ;       OCT 75  CORRECTED BUGS THAT CAUSED INCORRECT PRINTING
                                ;               WHEN THE LIST DEVICE WAS AN LV11.
                                ;
                                ;       AUG 75  INITIAL RELEASE
```

```
                        .SBTTL  MPG EXEC BUILD INFORMATION
                        .TITLE  MAINTDEC-11-DTECA-B  COMMON SECTION FOR ALL MPG EXECUTIVES

                        ;REVISION "B"

;*****************************************************************
; MACY11:
;
;    DTECA?,DTECA?/CRF:SYM/DOC=DTECA?.P11
;
;    OR FOR MIMIC,  DTECM?,DTECM?/CRF:SYM/EQ:MIMIC/DOC=DTECA?.P11
;
;      NOTE: "/EQ:NLIST" WILL SUPPRESS DISPLAY OF ALL
;            UNUSED CONDITIONAL CODE IN THE LISTING.
;
; LNKX11:
;
;    TC11 = TCMPG.BIN/B:0,TCMPG.MAP←DTECA?,DTE0A?/E
;    RK11 = RKMPG.BIN/B:0,RKMPG.MAP←DTECA?,DTE1A?/E
;    TM11 = TMMPG.BIN/B:0,TMMPG.MAP←DTECA?,DTE2A?/E
;    TM02 = THMPG.BIN/B:0,THMPG.MAP←DTECA?,DTE3A?/E
;    RX11 = RXMPG.BIN/B:0,RXMPG.MAP←DTECA?,DTE4A?/E
;    RP04 = RBMPG.BIN/B:0,RBMPG.MAP←DTECA?,DTE5A?/E
;
;    OR FOR UNDER MIMIC:
;
;    RK11 = RKMPG.BIN/B:0,RKMPG.MAP←DTECM?,DTE1A?,DTMGA?,XXXX/E
;
;          FOR RK11 THIS RESULTS IN A 578 BLOCK DISK.
;          WHERE XXXX IS AN OPTIONAL DEVICE ROUTINE.
;
;    THE LINK MAPS, WHICH ARE IN MAINDEC DOCUMENT #
;    11-DTEMA-?, PROVIDE THE ACTUAL MEMORY ADDRESSES
;    OF THE VARIOUS MODULES IN EACH VERSION OF THE
;    EXECUTIVE.
;
; FOR PAPER TAPE OUTPUT:
;
;    PUNCH TCMPG.BIN/FILE:ELEV
;    PUNCH RKMPG.BIN/FILE:ELEV
;    PUNCH TMMPG.BIN/FILE:ELEV
;    PUNCH THMPG.BIN/FILE:ELEV
;    PUNCH RXMPG.BIN/FILE:ELEV
;    PUNCH RBMPG.BIN/FILE:ELEV
;
; TCMPG  UNIQUE MODULES = MAINDEC # DTE0A-?
; RKMPG  UNIQUE MODULES = MAINDEC # DTE1A-?
; TMMPG  UNIQUE MODULES = MAINDEC # DTE2A-?
; THMPG  UNIQUE MODULES = MAINDEC # DTE3A-?
; RXMPG  UNIQUE MODULES = MAINDEC # DTE4A-?
; RBMPG  UNIQUE MODULES = MAINDEC # DTE5A-?
;
;*****************************************************************
```

```
                                .SBTTL  MPG EXEC GLOBALS AND EQUATES


                                .DSABL  GBL
                                .IIF DF NLIST,  .NLIST  CND

                                ;NOTE:  THE SIZE OF THE LINKER OUTPUT OF ANY VERSION OF THE
                                ;       EXECUTIVE CANNOT EXCEED OCTAL 20000 BYTES.


                                ;EXTERNAL GLOBAL NAMES

                                .GLOBL  CFTCHI,CSAVEI,DIRBLK,EXCID


                                ;INTERNAL GLOBAL NAMES

                                .GLOBL  BCDCV,BCLEAR,BKREAD,BLKWRT,BMOVE
                                .GLOBL  BUF,CHROUT,CLRBUF,CMPNAM,CRLF,CRPMAP
                                .GLOBL  DATUPK,DELAY,DELERR,DELFIL
                                .GLOBL  DEVERR,DEVFUL,EXECST
                                .GLOBL  FLMDTF,FSTMOD,ITOA,LDUNUM,LSTFLG,MAPBUF
                                .GLOBL  MBUF,MES,MMAP,NXTBLK
                                .GLOBL  PAKNAM,PIPFLG,READBK,RST04,SAV04
                                .GLOBL  TAB,TXEXT,TXNAM,UPKNAM,WRTBLK
                                .GLOBL  WRTLC,$BUF,$BUF10,$BUF2
                                .GLOBL  $MAPBF,$MBUF,$MMAP,$TPNM1,$TXNAM,$TYPNM


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
001200                          STACK   =1200


000020                          CTPRIO  =20             ;BIT DEF. FOR USER PROG FLGWD
000040                          SETDED  =40

000001                          MMVER   =1              ;BIT DEF. FOR MPG SYSTEM FLGWD
000040                          UNIMAP  =40
002000                          CUSPGR  =2000
004000                          CHPINT  =4000
010000                          CNTRLO  =10000
020000                          CTUDED  =20000
040000                          CRUDED  =40000
100000                          CTRQBP  =100000

177572                          MMMR0   =177572
172516                          MMMR3   =172516
```

```
                                        ;DEVICE DESCRIPTOR BLOCK EQUATES
                                        
                                        
177714                  XDHSU   =-64    ;INDEX TO DEV HANDLER SET UP
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    ;INDEX TO BLOCK # FOR MAP BLOCK
177726                  XSVCNT  =-52    ;INDEX TO ENTRY # IN UFD BLOCK
177730                  XSVBLK  =-50    ;INDEX TO BLK # OF UFD FOR CURR FILE
177732                  XSVNAM  =-46    ;INDEX TO FILE NAME IN RAD50 (2 WORDS)
177736                  XSVEXT  =-42    ;INDEX TO FILE EXTENSION NAME IN RAD50
177740                  XSVDAT  =-40    ;INDEX TO DATE IN DOS FORMAT
177742                  XSVXX   =-36    ;INDEX TO (NOT USED?)
177744                  XISTBK  =-34    ;INDEX TO BLK # OF FILE'S FIRST DATA BLOCK
177746                  XBKLGT  =-32    ;INDEX TO # OF BLOCKS IN THE FILE
177750                  XLSTBK  =-30    ;INDEX TO # OF LAST BLK WRITTEN
177752                  XSVUPT  =-26    ;INDEX TO (NOT USED?)
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     ;INDEX TO DRIVE NUMBER
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
```

```
                                .SBTTL  LOAD MEDIA BOOTSTRAP LOADER


000000'                         .CSECT  LOADER

                        ;  THE SELECTED DEVICE LOADER RESIDES IN THIS
                        ;  AREA BETWEEN OCTAL ADR'S 0 TO 577.


                        ;  THE LISTINGS THAT CONTAIN THE VARIOUS LOADERS WHICH
                        ;  MAY OCCUPY THIS AREA ARE:
                        ;
                        ;    TCMPG  = DTE0A-? - TC11 BOOT LOADER
                        ;    RKMPG  = DTE1A-? - RK11 BOOT LOADER
                        ;    TMMPG  = DTE2A-? - TM11 BOOT LOADER
                        ;    THMPG  = DTE3A-? - TM02 BOOT LOADER
                        ;    RXMPG  = DTE4A-? - RX11 BOOT LOADER
                        ;    RBMPG  = DTE5A-? - RP04 BOOT LOADER
```

```
                                        .SBTTL  HOUSEKEEPING EXTENSION & INITIAL ENTRY


        000000'                         .CSECT  EXEC


                                        ;INITIAL ENTRY AND HOUSEKEEPING EXTENSION

000000'                         LOC600:
            001                         .IF DF MIMIC
                                        JMP     DISKWR                  ;GO TO DISK CREATE
            000                         .ENDC
000000' 012706  001200          EXECST: MOV     #STACK,R6               ;SET UP STACK POINTER
000004' 004567  000016'                 JSR     R5,E1THKP               ;GO DO 1 TIME HOUSEKEEPING
            001                         .IF NDF MIMIC
000010' 016700  000456                  MOV     MPGST,R0                ;GET ADR OF WHERE MPG WILL GO
000014' 016701  000446                  MOV     ENDMEM,R1               ;GET END OF MEM ADR
000020' 160001                          SUB     R0,R1                   ;CALC AMNT OF MEM AFTER THE EXEC
000022' 010167  000022                  MOV     R1,EFMSZ                ;STORE IT IN LINK INFO
000026' 012720  046524                  MOV     #"TM,(R0)+              ;STORE MPG'S FILE NAME WHICH
000032' 012720                          MOV     (PC)+,(R0)+             ;IS 'TMGA??' OR 'TMMA??'
000034' 040507                  MPGID:  .ASCII  /GA/
000036' 012710  037477                  MOV     #"??,(R0)
000042' 004567  003176                  JSR     R5,LOADVR               ;GO LOAD MPG
000046' 000016'                 MPGADR: .WORD   MPGBGN                   ;  MEM ADR
000050' 000000                  EFMSZ:  .WORD   XXXX                     ;  MAX BYTE CNT
000052' 000060                          .WORD   LMPGDE-.                 ;  DEV ERR ADR
000054' 000050                          .WORD   LMPGIM-.                 ;  INSUF MEM ERR ADR
000056' 000002                          .WORD   LMPGIN-.                 ;  NORMAL RET ADR
            000                         .ENDC
000060' 012700                  LMPGIN: MOV     (PC)+,R0                ;GET STORED ADR OF INTERFACE ADR'S
000062' 000000                  ESGADR: .WORD   XXXX
000064' 016701  000402                  MOV     MPGST,R1                ;SET UP ADR IN MPG FOR THEM
000070' 062701  000006                  ADD     #6,R1
000074' 012702                          MOV     (PC)+,R2                ;GET # OF ADR'S STORED
000076' 000000                  ESGCNT: .WORD   XXXX
000100' 012021                  10$:    MOV     (R0)+,(R1)+             ;MOVE INTERFACE ADR TO MPG
000102' 005302                          DEC     R2                      ;DECR COUNT
000104' 001375                          BNE     10$                     ;DONE ALL? (Y,N-10$)
                                                                        ;R1 IS NOW POINTING AT MPG'S
                                                                        ;ENTRY POINT
000106' 012706  001200                  MOV     #STACK,R6               ;SET UP STACK ADR TO BE SURE
000112' 016700  000350                  MOV     ENDMEM,R0               ;SET UP END OF MEM ADR
000116' 012702                          MOV     (PC)+,R2                ;SET UP MPG'S POSITION ADJ FACTOR
000120' 000000                  POSADJ: .WORD   0                       ;  MAY BE CHANGED IF MEM MGMNT
000122' 000111                          JMP     (R1)                    ;GO TO MPG


000124' 004567  005524          LMPGIM: JSR     R5,MES                  ;ISSUE INSUFF MEM ERR MSG
000130' 000136'                         .WORD   ISUFM
000132' 000000                  LMPGDE: HALT                            ;ERROR DURING MPG LOAD
000134' 000721                          BR      EXECST                  ;GO TRY AGAIN

000136' 042452  025122  044440  ISUFM:  .ASCIZ  /*ER* INSUF MEM FOR MPG/<015><012>
000144' 051516  043125  046440
000152' 046505  043040  051117
```

```
                000160' 046440  043520  005015
                000166'    000
    338         000170'                        .EVEN
    339
    340
    341
    342                                        ;LIGHT THE LIGHTS S/R
    343
    344         000400'                        .=      LOC600+400
    345
    346 000400' 012701                 LIGHTS: MOV     (PC)+,R1        ;GET CURRENT BIT PATTERN
    347 000402' 000000                 LTSPAT: .WORD   0               ;BIT PATTERN STORAGE
    348 000404' 012702  000240                 MOV     #160.,R2        ;SET UP LOOP CNT
    349 000410' 050101                 10$:    BIS     R1,R1           ;BIT PATTERN TO THE LIGHTS
    350 000412' 050101                         BIS     R1,R1
    351 000414' 050101                         BIS     R1,R1
    352 000416' 005302                         DEC     R2              ;DECR LOOP CNT
    353 000420' 001373                         BNE     10$             ;CNT = 0? (Y,N-10$)
    354 000422' 005327                         DEC     (PC)+           ;DECR 100 X COUNT
    355 000424' 000144                 L2MCNT: .WORD   100.            ;100X CNT
    356 000426' 001016                         BNE     LTSEX           ;CNT = 0? (Y,N-LTSEX)
    357 000430' 012767  000144  177766         MOV     #100.,L2MCNT    ;RESTORE 100 X CNT
    358 000436' 005701                         TST     R1              ;BIT PATTERN FULL?
    359 000440' 100003                         BPL     20$             ;Y,N-20$
    360 000442' 005067  177734                 CLR     LTSPAT          ;INITIALIZE THE PATTERN
    361 000446' 000406                         BR      LTSEX           ;GO TO EXIT
    362 000450' 000261                 20$:    SEC                     ;EXPAND BIT PATTERN
    363 000452' 106067  177724                 RORB    LTSPAT
    364 000456' 000261                         SEC
    365 000460' 106167  177717                 ROLB    LTSPAT+1
    366 000464' 000207                 LTSEX:  RTS     PC              ;EXIT IN-LINE
```

```
                                        .SBTTL  MPG EXEC CONSTANTS & BUFFER AREAS

000466' 000000                  ENDMEM: .WORD   0                       ;LAST MEM WORD ADR STORAGE
000470' 000000                  TODAY:  .WORD   0                       ;TODAY'S DATE CODE
000472' 000016'                 MPGST:  .WORD   MPGBGN                  ;MPG'S MEM START ADR

        000000                  XXXX=   0                               ;DATA THAT WILL BE TAILORED

000474' 001144'                 $BUF:   .WORD   BUF                     ;ADDRESSES NEEDED BY HANDLERS
000476' 001146'                 $BUF2:  .WORD   BUF+2
000500' 001154'                 $BUF10: .WORD   BUF+10
000502' 000544'                 $MAPBF: .WORD   MAPBUF
000504' 000744'                 $MBUF:  .WORD   MBUF
000506' 000754'                 $MMAP:  .WORD   MMAP
000510' 000524'                 $TXNAM: .WORD   TXNAM

000512' 001200                  STKADR: .WORD   STACK
000514'    000                  LSTFLG: .BYTE   0                       ;DIRECTORY "ALL" FLAG
000515'    000                  FSTMOD: .BYTE   0                       ;DIRECTORY "FAST MODE" FLAG
000516'    000                  PRTFLG: .BYTE   0                       ;USE TTY OR PRNTR FLG
000517'    000                          .BYTE   0

000520' 000000                  CHKSUM: .WORD   0                       ;COMPUTED CKSUM WD STORAGE
000522'    000                  PIPFLG: .BYTE   0                       ;PIP MODE FLAG
000523'    000                          .BYTE   0
000524'    000      000     000 TXNAM:  .BYTE   0,0,0,0,0,0             ;ASCII FILENAME STORAGE
000527'    000      000     000
000532'    000      000     000 TXEXT:  .BYTE   0,0,0
000536'                            .EVEN

000536' 000006                  REALNM: .BLKB   6                       ;FILENAME OF FILE ON A MATCH

000544' 000100                  MAPBUF: .BLKW   64.                     ;MAP BUFFER

000744' 000004                  MBUF:   .BLKW   4                       ;ANOTHER MAP BUFFER
000754' 000074                  MMAP:   .BLKW   60.

001144'                         BUF:                                    ;MAIN READ - WRITE BUFFER

                                ;       THIS BUFFER AREA (256 WORDS) IS INITIALLY OCCUPIED BY ONE
                                ;       TIME HOUSEKEEPING S/R'S AND MESSAGES. THE FOLLOWING CODE AND
                                ;       MESSAGES UP TO 'BUFEND' WILL BE DESTROYED WHEN THE EXEC
                                ;       LOADS THE MPG PROGRAM FILE.
```

```
                                        .SBTTL  TEMPORARY ONE TIME HOUSEKEEPING S/R'S

                                ;***************
                                ;
                                ;  ASCII TO BINARY CONVERSION ROUTINE
                                ;
                                ;       JSR     PC,ATOI                 S/R CALL
                                ;
                                ;       USES VALUE AT "EKBPTR" AS INPUT DATA ADR
                                ;       R2 CONTAINS RESULT UPON EXIT
                                ;
                                ;***************

001144' 004467  003546  ATOI:   JSR     R4,SAV04                ;SAVE WORK REG'S
001150' 005002                  CLR     R2                      ;CLEAR RESULT REG
001152' 016700  001312'         MOV     EKBPTR,R0               ;GET ASCII DATA ADR
001156' 112003          10$:    MOVB    (R0)+,R3                ;GET ASCII CHAR
001160' 120327  000040          CMPB    R3,#40                  ;IS IT A SPACE?
001164' 001774                  BEQ     10$                     ;N,Y-10$
001166' 162703  000060  20$:    SUB     #'0,R3                  ;CONVERT IT TO BINARY
001172' 100412                  BMI     30$                     ;LESS THAN ASCII 0? (N,Y-30$)
001174' 120327  000011          CMPB    R3,#9.                  ;GREATER THAN ASCII 9?
001200' 101007                  BHI     30$                     ;N,Y-30$
001202' 006302                  ASL     R2                      ;MULT RUNNING TOTAL BY 10
001204' 060203                  ADD     R2,R3                   ;AND INCLUDE NEW DIGIT'S VALUE
001206' 006302                  ASL     R2
001210' 006302                  ASL     R2
001212' 060302                  ADD     R3,R2
001214' 112003                  MOVB    (R0)+,R3                ;GET NXT ASCII CHAR
001216' 000763                  BR      20$                     ;GO CHECK IT
001220' 005300          30$:    DEC     R0                      ;BACK UP DATA PNTR
001222' 010067  001312'         MOV     R0,EKBPTR               ;STORE IT
001226' 010266  000004          MOV     R2,4(SP)                ;STORE RESULT ON STACK
001232' 004767  003472          JSR     PC,RST04                ;RESTORE REG'S
001236' 000207                  RTS     PC                      ;EXIT IN-LINE


                                ;***************
                                ;
                                ;  READ AND CONVERT OCTAL ASCII TO BINARY S/R
                                ;
                                ;       JSR     PC,OATOB                S/R CALL
                                ;       BR      LABEL                   EXECUTED IF A DATA ERROR
                                ;
                                ;       ON EXIT:
                                ;       R0 = 0 = NO DATA RECEIVED
                                ;       R1 = DATA RECEIVED IN BINARY
                                ;
                                ;***************

001240' 012700  001314'  OATOB: MOV     #EKYBUF,R0              ;SET UP REL ADR OF
001244' 162700  001260'         SUB     #5$,R0                  ;KEYBOARD BUFFER
001250' 010067  000004          MOV     R0,5$
001254' 004567  005644          JSR     R5,CTRD                 ;ISSUE READ FOR DATA
001260' 000000          5$:     .WORD   XXXX
001262' 000025                  .WORD   21.
```

```
                001264' 005000                  CLR     R0              ;RESET DATA FND FLAG
                001266' 005001                  CLR     R1              ;CLEAR ACCUM REG
                001270' 012702  001314'         MOV     #EKYBUF,R2      ;INITIALIZE READ DATA PNTR
                001274' 112203          10$:    MOVB    (R2)+,R3        ;GET DATA CHAR
                001276' 122703  000040          CMPB    #40,R3          ;IS IT A SPACE?
                001302' 001774                  BEQ     10$             ;N,Y-10$
                001304' 122703  000015  20$:    CMPB    #015,R3         ;IS IT A C/R?
                001310' 001420                  BEQ     40$             ;N,Y-40$
                001312' 122703  000012          CMPB    #012,R3         ;IS IT A L/F?
                001316' 001415                  BEQ     40$             ;N,Y-40$
                001320' 162703  000060          SUB     #60,R3          ;MAKE CHAR A BINARY VALUE
                001324' 100420                  BMI     80$             ;LESS THAN ASCII 0? (N,Y-80$)
                001326' 020327  000007          CMP     R3,#07          ;GREATER THAN 7?
                001332' 101015                  BHI     80$             ;N,Y-80$
                001334' 005200                  INC     R0              ;SET DATA FND FLAG
                001336' 006301                  ASL     R1              ;MULT ANY PREV CHAR'S BY 8
                001340' 006301                  ASL     R1
                001342' 006301                  ASL     R1
                001344' 060301                  ADD     R3,R1           ;ADD IN CURRENT CHAR
                001346' 112203                  MOVB    (R2)+,R3        ;GET NEXT CHAR
                001350' 000755                  BR      20$             ;GO PROCESS IT
                001352' 020127  000377  40$:    CMP     R1,#377         ;VALUE TOO LARGE?
                001356' 101003                  BHI     80$             ;N,Y-80$
                001360' 062716  000002          ADD     #2,(SP)         ;SET UP NORMAL RET POINT
                001364' 000207          50$:    RTS     PC              ;EXIT
                001366' 004567  004262  80$:    JSR     R5,MES          ;ISSUE ERROR MSG
                001372' 001725'                 .WORD   INVDMG
                001374' 000773                  BR      50$             ;RETURN FOR ERROR EXIT
                    001                         .IF DF MIMIC


                                        ;***************
                                        ;
                                        ;  OCTAL BINARY TO ASCII S/R
                                        ;
                                        ;       JSR     PC,OBTOA                S/R CALL
                                        ;
                                        ;       R1 = BINARY WORD TO CONVERT
                                        ;       R2 = ADR TO STORE 6 ASCII DIGITS
                                        ;       DESTROYS R0,R1,R2,R3
                                        ;
                                        ;***************

                                OBTOA:  CLR     R0              ;CLEAR ACCUM REG
                                        MOV     #6,R3           ;SET UP DIGIT CNT
                                        BR      20$             ;GO DO BIT 15
                                10$:    CLR     R0              ;CLEAR ACCUM REG
                                        ROL     R1              ;SHIFT 3 BITS INTO R0
                                        ROL     R0
                                        ROL     R1
                                        ROL     R0
                                20$:    ROL     R1
                                        ROL     R0
                                        BIS     #60,R0          ;MAKE CHAR ASCII
                                        MOVB    R0,(R2)+        ;STORE IT IN OUTPUT AREA
                                        DEC     R3              ;DECR DIGIT CNT
```

```
                                        BNE     10$                     ;DONE ALL? (Y,N-10$)
                                        RTS     PC                      ;EXIT IN-LINE
                000                     .ENDC


                                ;***************
                                ;
                                ;  CHECK KEYBOARD REPLY S/R
                                ;
                                ;       JSR     PC,CKRPLY       S/R CALL
                                ;       BR      LABEL           EXECUTED IF INV DATA
                                ;       BR      LABEL           EXECUTED IF Y, CR, OR LF
                                ;                               IN-LINE IF N
                                ;
                                ;***************

001376' 012702  001314'         CKRPLY: MOV     #EKYBUF,R2              ;INITIALIZE READ DATA POINTER
001402' 112203                  10$:    MOVB    (R2)+,R3                ;GET DATA BYTE
001404' 122703  000040                  CMPB    #40,R3                  ;IS IT A SPACE?
001410' 001774                          BEQ     10$                     ;N,Y-10$
001412' 122703  000015                  CMPB    #015,R3                 ;IS IT A C/R?
001416' 001413                          BEQ     20$                     ;N,Y-20$
001420' 122703  000012                  CMPB    #012,R3                 ;IS IT A L/F?
001424' 001410                          BEQ     20$                     ;N,Y-20$
001426' 122703  000131                  CMPB    #'Y,R3                  ;IS IT A "Y"?
001432' 001405                          BEQ     20$                     ;N,Y-20$
001434' 122703  000116                  CMPB    #'N,R3                  ;IS IT AN "N"?
001440' 001004                          BNE     30$                     ;Y,N-30$
001442' 062716  000002                  ADD     #2,(SP)                 ;BYPASS SECOND BR
001446' 062716  000002          20$:    ADD     #2,(SP)                 ;BYPASS FIRST BR
001452' 000207                  30$:    RTS     PC                      ;EXIT
```

```
                                                .SBTTL  TEMPORARY ONE TIME HOUSEKEEPING MESSAGE STORAGE AREA

001454' 005015  042077  052101  EDMSG:  .ASCIZ  <015><012>/?DATE  (DD-MMM-YY)/<015><012>/*/
001462' 020105  024040  042104
001470' 046455  046515  054455
001476' 024531  005015  000052
001504' 042452  025122  044440  EDERMG: .ASCIZ  /*ER* INV DATE/
001512' 053116  042040  052101
001520' 000105
001522' 005015  047503  051516  CON1MG: .ASCII  <015><012>'CONS TERMINAL CONSTANTS:'<015><012><015><012>
001530' 052040  051105  044515
001536' 040516  020114  047503
001544' 051516  040524  052116
001552' 035123  005015  005015
001560' 043077  046111  020114          .ASCIZ  '?FILL AFTER:  012 / '
001566' 043101  042524  035122
001574' 020040  030460  020062
001602' 020057     000
001605'    077  044506  046114  CON2MG: .ASCIZ  '?FILL WITH:   000 / '
001612' 053440  052111  035110
001620' 020040  030040  030060
001626' 027440  000040
001632' 043077  046111  020114  CON3MG: .ASCIZ  '?FILL COUNT:  002 / '
001640' 047503  047125  035124
001646' 020040  030060  020062
001654' 020057     000
001657'    077  047503  053116  CON4MG: .ASCIZ  '?CONVERT LOWER CASE TO UPPER (Y/N) / '
001664' 051105  020124  047514
001672' 042527  020122  040503
001700' 042523  052040  020117
001706' 050125  042520  020122
001714' 054450  047057  020051
001722' 020057     000
001725'    052  051105  020052  INVDMG: .ASCIZ  '*ER* INV DATA'<015><012>
001732' 047111  020126  040504
001740' 040524  005015     000
001745'    015  037412  047514  LDUREQ: .ASCIZ  <015><012>'?LOAD/SAVE/FETCH DEV UNIT # / '
001752' 042101  051457  053101
001760' 027505  042506  041524
001766' 020110  042504  020126
001774' 047125  052111  021440
002002' 027440  000040
002006' 005015  052477  042523  USEMM:  .ASCIZ  <015><012>'?USE MEM MGMNT (Y/N) / '
002014' 046440  046505  046440
002022' 046507  052116  024040
002030' 027531  024516  027440
002036' 000040
002040' 042440  042530  052503  EXCIDT: .ASCIZ  ' EXECUTIVE FOR MPG'<015><012>
002046' 044524  042526  043040
002054' 051117  046440  043520
002062' 005015     000
        002066'                         .EVEN
        002144'                         .=      BUF+512.
002144' 000000                  BUFEND: .WORD   0
```

```
                                        .SBTTL  INPUT DEVICE DESCRIPTOR BLOCK (DDB)

002146' 000000                  IDHSU:  .WORD   0       ;XDHSU    DEVICE HANDLER SET UP ADDRESS
002150'                         DDBSTR:
002150' 000000                  IWCTR:  .WORD   0       ;XWCTR    OUTPUT FILE OPEN FLAG
002152' 000000                  IFLMOD: .WORD   0       ;XFLMOD   FILE MODE FLAG
002154' 000000                  IFLCNT: .WORD   0       ;XFLCNT   FILE COUNT
002156' 000000                  ISVMAP: .WORD   0       ;XSVMAP   BLK # OF FILE'S MAP BLOCK
002160' 000000                  ISVCNT: .WORD   0       ;XSVCNT   ENTRY # OF CURR FILE IN UFD
002162' 000000                  ISVBLK: .WORD   0       ;XSVBLK   BLK # OF CURR FILE'S UFD
                                                        ;UFD DIRECTORY ENTRY DATA FOR FILE
002164' 000000  000000          ISVNAM: .WORD   0,0     ;XSVNAM   FILE'S FILENAME IN RAD50 (2 WORDS)
002170' 000000                  ISVEXT: .WORD   0       ;XSVEXT   FILE'S EXTENSION IN RAD50
002172' 000000                  ISVDAT: .WORD   0       ;XSVDAT   FILE'S CREATION DATE IN DOS FORMAT
002174' 000000                  ISVXX:  .WORD   0       ;XSVXX    (NOT USED?)
002176' 000000                  IISTBK: .WORD   0       ;XISTBK   BLOCK # OF FILE'S FIRST DATA BLK
002200' 000000                  IBKLGT: .WORD   0       ;XBKLGT   # OF BLOCKS IN THE FILE
002202' 000000                  ILSTBK: .WORD   0       ;XLSTBK   BLOCK # OF LAST DATA BLOCK WRITTEN
002204' 000000                  ISVUPT: .WORD   0       ;XSVUPT   (NOT USED?)
                                                        ;END OF DIRECTORY ENTRIES
002206' 000000                  INBOOT: .WORD   0       ;XBT      ADDRESS OF "BOOT" ROUTINE
002210' 000000                  INDRT:  .WORD   0       ;DRT      ADDRESS OF "DIRECTORY" ROUTINE
002212' 000000                  INZER:  .WORD   0       ;ZER      ADDRESS OF "ZERO" ROUTINE
002214' 000000                  INDLT:  .WORD   0       ;DLT      ADDRESS OF "DELETE" ROUTINE
002216' 000000                  INCLS:  .WORD   0       ;CLS      ADDRESS OF "CLOSE" ROUTINE
002220' 000000                  INETR:  .WORD   0       ;ETR      ADDRESS OF "ENTER" (CREATE) ROUTINE
002222' 000000                  INSRH:  .WORD   0       ;SRH      ADDRESS OF "LOOKUP" (SEARCH) ROUTINE
002224' 000000                  INALC:  .WORD   0       ;ALC      ADDRESS OF "ALLOCATE" ROUTINE
002226' 000000                  INSRV:  .WORD   0       ;XSV      ADDRESS OF DEVICE DRIVER ROUTINE
002230' 000000                  INDRV:  .WORD   0       ;XDN      CURRENT DRIVE (UNIT) NUMBER
002232'                         INDEV:                  ;R5 POINTS HERE
002232' 000000                  INCM:   .WORD   0       ;XCM      ADDRESS OF DEVICE'S COMMAND REGISTER
002234' 000000                  INWC:   .WORD   0       ;XWC      CURRENT WORD COUNT
002236' 000000                  INBA:   .WORD   0       ;XBA      CURRENT BUS (MEMORY) ADDRESS
002240'                         INBLK:
002240' 000000                  INDT:   .WORD   0       ;XDT      CURRENT BLOCK NUMBER
002242' 000000                  INCOM:  .WORD   0       ;XCO      CURRENT COMMAND CODE
002244' 000000                  INPRC:  .WORD   0       ;XRD      READ COMMAND CODE
002246' 000000                  INPWC:  .WORD   0       ;XWT      WRITE COMMAND CODE
002250' 000000                  INBKCT: .WORD   0       ;XBC      REQUESTED BLOCK COUNT
002252' 000000                  INDIR:  .WORD   0       ;XDR      ADDRESS OF FIRST DIRECTORY BLK #
002254' 000000                  INNBK:  .WORD   0       ;XNB      LAST BLOCK # ALLOCATED (NEXT BLK #)
002256'    000  000  000        IFNAM:  .BYTE   0,0,0   ;XXNAM    FILE'S NAME IN ASCII (9 CHAR'S)
002261'    000  000  000                .BYTE   0,0,0
002264'    000  000  000                .BYTE   0,0,0
        002270'                         .EVEN
        002270'                 INDEND= .
```

```
                                .SBTTL  OUTPUT DEVICE DESCRIPTOR BLOCK (DDB).

002270' 000000          ODHSU:  .WORD   0       ;XDHSU    DEVICE HANDLER SET UP ADDRESS
002272' 000000          OWCTR:  .WORD   0       ;XWCTR    OUTPUT FILE OPEN FLAG
002274' 000000          OFLMOD: .WORD   0       ;XFLMOD   FILE MODE FLAG
002276' 000000          OFLCNT: .WORD   0       ;XFLCNT   FILE COUNT
002300' 000000          OSVMAP: .WORD   0       ;XSVMAP   BLK # OF FILE'S MAP BLOCK
002302' 000000          OSVCNT: .WORD   0       ;XSVCNT   ENTRY # OF CURR FILE IN UFD
002304' 000000          OSVBLK: .WORD   0       ;XSVBLK   BLK # OF CURR FILE'S UFD
                                                ;UFD DIRECTORY ENTRY DATA FOR FILE
002306' 000000 000000   OSVNAM: .WORD   0,0     ;XSVNAM   FILE'S FILENAME IN RAD50 (2 WORDS)
002312' 000000          OSVEXT: .WORD   0       ;XSVEXT   FILE'S EXTENSION IN RAD50
002314' 000000          OSVDAT: .WORD   0       ;XSVDAT   FILE'S CREATION DATE IN DOS FORMAT
002316' 000000          OSVXX:  .WORD   0       ;XSVXX    (NOT USED?)
002320' 000000          O1STBK: .WORD   0       ;X1STBK   BLOCK # OF FILE'S FIRST DATA BLK
002322' 000000          OBKLGT: .WORD   0       ;XBKLGT   # OF BLOCKS IN THE FILE
002324' 000000          OLSTBK: .WORD   0       ;XLSTBK   BLOCK # OF LAST DATA BLOCK WRITTEN
002326' 000000          OSVUPT: .WORD   0       ;XSVUPT   (NOT USED?)
                                                ;END OF DIRECTORY ENTRIES
002330' 000000          OUBOOT: .WORD   0       ;XBT      ADDRESS OF "BOOT" ROUTINE
002332' 000000          OUDRT:  .WORD   0       ;DRT      ADDRESS OF "DIRECTORY" ROUTINE
002334' 000000          OUZER:  .WORD   0       ;ZER      ADDRESS OF "ZERO" ROUTINE
002336' 000000          OUDLT:  .WORD   0       ;DLT      ADDRESS OF "DELETE" ROUTINE
002340' 000000          OUCLS:  .WORD   0       ;CLS      ADDRESS OF "CLOSE" ROUTINE
002342' 000000          OUETR:  .WORD   0       ;ETR      ADDRESS OF "ENTER" (CREATE) ROUTINE
002344' 000000          OUSRH:  .WORD   0       ;SRH      ADDRESS OF "LOOKUP" (SEARCH) ROUTINE
002346' 000000          OUALC:  .WORD   0       ;ALC      ADDRESS OF "ALLOCATE" ROUTINE
002350' 000000          OUTSRV: .WORD   0       ;XSV      ADDRESS OF DEVICE DRIVER ROUTINE
002352' 000000          OUTDRV: .WORD   0       ;XDN      CURRENT DRIVE (UNIT) NUMBER
002354'                 OUTDEV:                 ;RS POINTS HERE
002354' 000000          OUTCM:  .WORD   0       ;XCM      ADDRESS OF DEVICE'S COMMAND REGISTER
002356' 000000          OUTWC:  .WORD   0       ;XWC      CURRENT WORD COUNT
002360' 000000          OUTBA:  .WORD   0       ;XBA      CURRENT BUS (MEMORY) ADDRESS
002362'                 OUTBLK:
002362' 000000          OUTDT:  .WORD   0       ;XDT      CURRENT BLOCK NUMBER
002364' 000000          OUTCOM: .WORD   0       ;XCO      CURRENT COMMAND CODE
002366' 000000          OUPRC:  .WORD   0       ;XRD      READ COMMAND CODE
002370' 000000          OUPWC:  .WORD   0       ;XWT      WRITE COMMAND CODE
002372' 000000          OUBKCT: .WORD   0       ;XBC      REQUESTED BLOCK COUNT
002374' 000000          OUTDIR: .WORD   0       ;XDR      ADDRESS OF FIRST DIRECTORY BLK #
002376' 000000          OUTNBK: .WORD   0       ;XNB      LAST BLOCK # ALLOCATED (NEXT BLK #)
002400'    000    000   000    OFNAM:  .BYTE   0,0,0    ;XXNAM    FILE'S NAME IN ASCII (9 CHAR'S)
002403'    000    000   000            .BYTE   0,0,0
002406'    000    000   000            .BYTE   0,0,0
        002412'                        .EVEN
002412'                         DDBEND:
```

```
                                        .SBTTL  FILE ROUTINES REQUIRED FOR MPG


                                        ;512 BYTE DATA BLOCK FORMAT

                                                ;NEXT BLOCK # (NOT USED FOR PAPER TAPE) - 1 WORD
                                                ;SYNC WORD (WORD OF 000001) - 1 WORD
                                                ;BYTE COUNT (# OF DATA BYTES + 6) - 1 WORD
                                                ;REL MEMORY ADR FOR DATA - 1 WORD
                                                ;DATA BYTES - MAX OF 492.
                                                ;CHECKSUM BYTE - 1 BYTE
                                                ;UNUSED BYTES - 2 WORDS (GAP IF PAPER TAPE)
                                                ;UNUSED BYTES - 7 BYTES (WILL CONTAIN THE "EOF" BLK
                                                ;                       IF LAST BLOCK OF THE FILE.)


                                        ;READ LINKED FILE BLOCK

                                                ;JSR     R5,GET          S/R CALL
                                                ;.WORD   ERRADR-.        REL ERROR RET ADR
                                                ;.WORD   XXXX            ABS DATA START ADR ON RETURN
                                                ;.WORD   XXXX            DATA BYTE COUNT ON RETURN

002412' 004567  001554          GET:    JSR     R5,CREGSV               ;SAVE REG'S R0-R5
002416' 105767  000362                  TSTB    EPARBK                  ;PARTIAL BLOCK FLAG SET?
002422' 001116                          BNE     85$                     ;N,Y-85$
002424' 004767  001750          10$:    JSR     PC,ERDLNK               ;READ A  LINKED BLOCK
002430' 000551                          BR      EGERET                  ;ERROR ON READ? (N,Y-EXERET)
002432' 012700  001146'                 MOV     #BUF+2,R0               ;SET UP BUFFER ADR
002436' 005067  000340          20$:    CLR     EUSRCT                  ;CLEAR USER DATA BYTE COUNT
002442' 005067  176052                  CLR     CHKSUM                  ;INITIALIZE CHECKSUM VALUE
002446' 004767  000270                  JSR     PC,EGETBY               ;GET A BYTE (WILL BE IN R3)
002452' 005303                          DEC     R3                      ;IS IT THE 1 OF A POSSIBLE SYNC WORD?
002454' 001370                          BNE     20$                     ;Y,N-20$
002456' 004767  000260                  JSR     PC,EGETBY               ;GET ANOTHER BYTE (SYNC IS WORD OF 1)
002462' 105703                          TSTB    R3                      ;IS IT THE 0 OF A SYNC WD?
002464' 001364                          BNE     20$                     ;Y,N-20$
002466' 004767  000222                  JSR     PC,EGETWD               ;GET DATA BYTE COUNT
002472' 162703  000006                  SUB     #6,R3                   ;ALLOW FOR THE 3 HEADER WORDS
002476' 010367  000270                  MOV     R3,EDTACT               ;SAVE DATA COUNT
002502' 001002                          BNE     30$                     ;THIS THE EOF BLK? (Y,N-30$)
002504' 105267  000275                  INCB    ERDEOF                  ;SET THE EOF BLK FLAG
002510' 004767  000200          30$:    JSR     PC,EGETWD               ;GET MEM ADR/START ADR WORD
002514' 010367  000254                  MOV     R3,EMSADR               ;SAVE IT
002520' 010067  000254          40$:    MOV     R0,EUSRSA               ;SAVE BUF POINTER AS USER START ADR
002524' 112003                  50$:    MOVB    (R0)+,R3                ;GET A DATA BYTE
002526' 060367  175766                  ADD     R3,CHKSUM               ;INCLUDE IT IN THE CKSUM
002532' 005367  000234                  DEC     EDTACT                  ;DECR BLOCK DATA COUNT
002536' 100021                          BPL     65$                     ;WAS IT CKSUM BYTE? (Y,N-65$)
002540' 105767  175754                  TSTB    CHKSUM                  ;IS THE CHECK SUM CORRECT?
002544' 001402                          BEQ     55$                     ;N,Y-55$
002546' 000167  003672                  JMP     CKSMER                  ;GO REPORT CKECK SUM ERROR
002552' 105767  000227          55$:    TSTB    ERDEOF                  ;IS THIS THE EOF BLK?
002556' 001026                          BNE     80$                     ;N,Y-80$
002560' 020027  002144'                 CMP     R0,#BUF+512.            ;AT THE END OF THE BUFFER?
002564' 001003                          BNE     60$                     ;Y,N-60$
```

```
                                                CLRB    EPARBK          ;RESET PARTIAL BLOCK FLG
                                                BR      75$             ;GO TO EXIT
                                        60$:    MOV     R0,ENXDTA       ;SAVE CURR POINTER AS NEXT DATA ADR
                                                BR      70$             ;GO CHECK FOR EXIT
                                        65$:    INC     EUSRCT          ;ADD 1 TO USER DATA COUNT
                                                CMP     R0,#BUF+512.    ;AT THE END OF THE BUFFER?
                                                BNE     50$             ;Y,N-50$
                                                MOV     #BUF+2,ENXDTA   ;SET NXT DATA ADR TO START  OF BUF
                                        70$:    INCB    EPARBK          ;SET  PARTIAL  BLOCK FLAG
                                        75$:    TST     EUSRCT          ;USER DATA CNT = 0?
                                                BEQ     85$             ;N,Y-85$
                                        80$:    MOV     STKADR,R6       ;RESTORE STACK POINTER
                                                JSR     R5,CRTREG       ;RESTORE REGISTERS
                                                TST     (R5)+           ;POINT PAST ERR RET ADR
                                                MOV     EUSRSA,(R5)+    ;STORE DATA START ADR IN LINK INFO
                                                MOV     EUSRCT,(R5)+    ;DO SAME WITH DATA COUNT
                                                RTS     R5              ;RETURN TO THE USER

                                        85$:    MOV     ENXDTA,R0       ;GET NEXT DATA ADR
                                                CMP     R0,#BUF+2       ;IS IT AT THE START OF THE BUF?
                                                BNE     20$             ;Y,N-20$
                                                JSR     PC,ERDLNK       ;READ NEXT LINKED BLOCK
                                                BR      EXERET          ;ERROR ON READ? (N,Y-EXERET)
                                                ADD     EUSRCT,EMSADR   ;ADD PREV DATA CNT TO MEM START ADR
                                                CLR     EUSRCT          ;CLEAR USER'S DATA BYTE COUNT
                                                BR      40$             ;GO PROCESS USER DATA


                                        EGETWD: JSR     PC,EGETBY       ;GET A BYTE
                                                MOVB    R3,(PC)+        ;STORE IT
                                        ETMP:   .WORD   0               ;WORK WORD
                                                JSR     PC,EGETBY       ;GET NEXT BYTE
                                                MOVB    R3,ETMP+1       ;STORE IT ALSO
                                                MOV     ETMP,R3         ;GET BOTH STORED BYTES
                                                RTS     PC              ;EXIT IN-LINE

                                        EGETBY: CMP     R0,#BUF+512.    ;EXHAUSTED DATA BLOCK?
                                                BNE     EGETKG          ;Y,N-EGETKG
                                                JSR     PC,ERDLNK       ;READ NEXT BLOCK
                                        EGERET: BR      EXERET          ;ERROR ON READ? (N,Y-EXERET)
                                                MOV     #BUF+2,R0       ;RESTORE DATA ADR
                                        EGETKG: MOVB    (R0)+,R3        ;GET BYTE FROM BLOCK
                                                ADD     R3,CHKSUM       ;ADD IT INTO CHECK SUM
                                                RTS     PC              ;EXIT IN-LINE


                                        EDTACT: .WORD   0               ;COUNT OF DATA BYTES IN BLOCK
                                        EMSADR: .WORD   0               ;MEMORY ADR OR START ADR
                                        ENXDTA: .WORD   0               ;NEXT DATA ADR
                                        EUSRSA: .WORD   0               ;USER'S DATA START ADR
                                        EUSRCT: .WORD   0               ;USER'S DATA BYTE COUNT
                                        EPARBK: .BYTE   0               ;PARTIAL BLOCK FLAG
                                        ERDEOF: .BYTE   0               ;PROCESSING EOF BLK FLAG
```

```
                                        ;OPEN LINKED INPUT FILE
                                        ;
                                        ;       JSR     R5,OPENL        S/R CALL
                                        ;       .WORD   ERRADR-.        RELATIVE ERROR RETURN ADR
                                        ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
                                        ;
003006' 004767  001040          OPENL:  JSR     PC,ESUFNM               ;SET UP ".USR" EXT & FTCH I.D.
003012' 004567  001154          LDROPN: JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003016' 005067  177762                  CLR     EPARBK                  ;RESET PARTIAL BLK & EOF BLK FLAGS
003022' 004767  001006                  JSR     PC,ESUFT1               ;DO COMMON SET UP FUNCTIONS
003026' 012765  000001  177744          MOV     #1,XISTBK(R5)           ;SET UP DUMMY BLK #
003034' 012767  002256' 003322          MOV     #IFNAM,NFNAME           ;INIT FILENAME ADR
003042' 004775  177770                  JSR     PC,@SRH(R5)             ;GO TO DEV HDLR & LOOK FOR FILE
003046' 000417                          BR      EJNFND                  ;FILE FOUND? (Y,N-FLNOTF)
003050' 016565  177744  000006          MOV     XISTBK(R5),XDT(R5)      ;MOVE 1ST BLK # TO CURR BLK #
003056' 016706  175430          EXNRET: MOV     STKADR,R6               ;RESTORE STACK POINTER
003062' 004567  005752                  JSR     R5,CRTREG               ;RESTORE REGISTERS
003066' 005725                  EXDRET: TST     (R5)+                   ;INCR PAST ERROR RET ADR
003070' 000205                          RTS     R5                      ;EXIT TO MPG
                                        ;
003072' 016706  175414          EXERET: MOV     STKADR,R6               ;RESTORE STACK POINTER
003076' 004567  005736                  JSR     R5,CRTREG               ;RESTORE REGISTERS
003102' 061505                          ADD     (R5),R5                 ;SET UP ERROR RETURN ADR
003104' 000205                          RTS     R5                      ;EXIT TO MPG ERROR ADR
                                        ;
003106' 000167  003244          EJNFND: JMP     FLNOTF                  ;GO TO NOT FOUND ERROR
                                        ;
                                        ;
                                        ;CREATE OUTPUT LINKED FILE
                                        ;
                                        ;       JSR     R5,CREATE       S/R CALL
                                        ;       .WORD   NNNN            # OF BLOCKS NEEDED
                                        ;       .WORD   ERRADR-.        REL ERR RET ADR
                                        ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"
                                        ;
003112' 005725                  CREATE: TST     (R5)+                   ;BYPASS # OF BLOCKS
003114' 004567  001052                  JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003120' 004767  001700                  JSR     PC,CLRBUF               ;CLEAR BUFFER AREA
003124' 004767  000632                  JSR     PC,ESUSAV               ;DO COMMON SETUP FUNCTIONS
003130' 004767  001156                  JSR     PC,EINTOL               ;INITIALIZE FOR OUTPUT LINKED FILE
003134' 000750                          BR      EXNRET                  ;EXIT TO MPG VIA NORMAL RETURN
                                        ;
                                        ;WRITE LINKED ABS. FORMAT BLOCK
                                        ;CLOSE LINKED ABS. FORMAT FILE
                                        ;
                                        ;       JSR     R5,PUT          PUT S/R CALL
                                        ;               ,CLOSE          CLOSE S/R CALL
                                        ;       .WORD   ERRADR-.        RELERR RET ADR
                                        ;
003136' 005067  176002          CLOSE:  CLR     BUF                     ;INDICATE THIS IS THE LAST BLOCK
003142' 004567  001024          PUT:    JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003146' 005003                          CLR     R3                      ;INITIALIZE CHECKSUM VALUE
003150' 012700  001146'                 MOV     #BUF+2,R0               ;GET BUFFER START ADR
003154' 016002  000002                  MOV     2(R0),R2                ;GET BYTE COUNT
003160' 012710  000001                  MOV     #1,(R0)                 ;PLACE SYNC WORD IN BLOCK
```

```
                                10$:    MOVB    (R0)+,R4        ;GET BYTE OF BLOCK
                                        ADD     R4,R3           ;ADD IT INTO CKSUM
                                        DEC     R2              ;DECR BYTE COUNT
                                        BNE     10$             ;CNT = 0? (Y,N-10$)
                                        NEGB    R3              ;NEGATE THE CKSUM
                                        MOVB    R3,(R0)+        ;STORE CKSUM IN THE BLOCK
                                        TST     BUF             ;THIS THE LAST BLOCK?
                                        BNE     20$             ;Y,N-20$
                                        ADD     #4,R0           ;ALLOW FOR 4 BLANKS
                                        MOVB    #1,(R0)+        ;STORE EOF BLOCK DATA
                                        CLRB    (R0)+           ;  SYNC WORD
                                        MOVB    #6,(R0)+        ;  CNT OF 6
                                        CLRB    (R0)+
                                        CLRB    (R0)+           ;  START ADR OF 0
                                        CLRB    (R0)+
                                        MOVB    #371,(R0)+      ;  CKSUM
                                20$:    JSR     PC,EWRITL       ;GO WRITE LINKED BLOCK
                                        BR      EXNRET          ;EXIT TO MPG VIA NORMAL RETURN


                                ;LOAD DEVICE ROUTINE

                                        ;ALSO USED TO LOAD THE MPG PROGRAM FILE AND
                                        ;THE VALID DEVICES FILE

                                        ;JSR    R5,LOADVR       S/R CALL
                                        ;.WORD  MEMADR          ABS STARTING MEM ADR FOR DATA
                                        ;.WORD  BYTCNT          SIZE OF MEM AREA IN BYTES
                                        ;.WORD  DEVERR-.        REL RET ADR IF DEVICE ERR
                                        ;.WORD  INSFADR-.       REL RET ADR IF INSUFF MEM
                                        ;.WORD  NORMADR-.       REL NORMAL RET ADR
                                        ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"

                                LOADVR: MOV     (R5)+,(PC)+     ;GET STARTING MEM ADR
                                LDRMST: .WORD   0               ;STARTING & CURR MEM ADR
                                        MOV     (R5)+,(PC)+     ;GET BYTE COUNT
                                LDRBYC: .WORD   0               ;MEM AREA BYTE COUNT
                                        JSR     R5,CREGSV       ;SAVE ALL REG'S
                                        MOV     #CFTCHI+4,R0    ;GET FETCH INFO ADR
                                        MOV     (R0)+,-(SP)     ;SAVE FETCH'S MDL CODE &
                                        MOV     (R0),-(SP)      ;UNIT #
                                        MOV     -4(R0),(R0)     ;SET FETCH TO THE LOAD DEV
                                        MOV     -6(R0),-(R0)
                                        MOV     STKADR,-(SP)    ;SAVE STORED STACK POINTER
                                        MOV     #"MP,IFNAM+6    ;SET UP ".MPG" FILENAME EXT
                                        MOVB    #'G,IFNAM+8.
                                        MOV     #"LO,EFDEVN     ;SET FUNC DEV NAME TO "LOAD"
                                        MOV     #"AD,EFDEVN+2
                                        JSR     R5,LDROPN       ;GO OPEN DEVICE ROUT FILE
                                        .WORD   LDEVER-.
                                        CLR     @LDRMST         ;CLEAR FIRST WORD OF DEV ROUT AREA
                                LDRGET: JSR     R5,GET          ;GET A BLOCK OF THE FILE
                                        .WORD   LDEVER-.
                                LDRDTA: .WORD   XXXX            ;    ADR OF DATA IN BUF
                                LDRDTC: .WORD   XXXX            ;    # OF DATA BYTES IN BUF
                                        MOV     LDRDTA,R0       ;GET ADR OF DATA IN BUF
```

```
003364' 016702  177766                   MOV     LDRDTC,R2               ;GET # OF DATA BYTES IN BUF
003370' 001415                           BEQ     LDRNEX                  ;EOF BLK? (N,Y-LDRNEX)
003372' 016701  177650                   MOV     LDRMST,R1               ;GET CURR MEM ADR FOR DATA
003376' 005767  177650          20$:     TST     LDRBYC                  ;MEMORY AREA FILLED?
003402' 001440                           BEQ     LDRISM                  ;N,Y-LDRISM
003404' 112021                           MOVB    (R0)+,(R1)+             ;MOVE DATA BYTE TO DESIGNATED AREA
003406' 005367  177640                   DEC     LDRBYC                  ;DECR MEMORY AREA BYTE CNT
003412' 005302                           DEC     R2                      ;DECR BUF DATA BYTE CNT
003414' 001370                           BNE     20$                     ;CNT = 0?  (Y,N-20$)
003416' 010167  177624                   MOV     R1,LDRMST               ;STORE NEW CURR MEM ADR
003422' 000751                           BR      LDRGET                  ;GO GET NEXT FILE BLOCK
003424' 012767  000003  177270  LDRNEX:  MOV     #3,ETMP                 ;SET EXIT ADR TO NORMAL RET
003432' 012667  175054          LDREXC:  MOV     (SP)+,STKADR            ;GET ORG STACK POINTER
003436' 012667  000006G                  MOV     (SP)+,CFTCHI+6          ;RESTORE FETCH UNIT # &
003442' 012667  000004G                  MOV     (SP)+,CFTCHI+4          ;MDL CODE
003446' 016706  175040                   MOV     STKADR,R6               ;RESTORE STACK POINTER
003452' 004567  005362                   JSR     R5,CRTREG               ;RESTORE REGISTERS
003456' 005367  177240          30$:     DEC     ETMP                    ;DECR EXIT RETURN ADR CNT
003462' 001402                           BEQ     40$                     ;POINTING AT CORR ADR?  (N,Y-40$)
003464' 005725                           TST     (R5)+                   ;INCR PAST CURR ADR
003466' 000773                           BR      30$                     ;GO CK IF THIS IS THE ADR
003470' 061505                  40$:     ADD     (R5),R5                 ;SET UP RETURN ADR
003472' 000205                           RTS     R5                      ;RETURN TO CALLER

003474' 012767  000001  177220  LDEVER:  MOV     #1,ETMP                 ;SET RET ADR TO DEV ERROR ADR
003502' 000753                           BR      LDREXC                  ;GO TO COMMON EXIT
003504' 012767  000002  177210  LDRISM:  MOV     #2,ETMP                 ;SET RET ADR TO INSUFF MEM ADR
003512' 000747                           BR      LDREXC                  ;GO TO COMMON EXIT


                                         ;DELETE LINKED ".USR" FILE

                                                 ;JSR    R5,DELETE       S/R CALL
                                                 ;.WORD  ERRADR-.        REL ERR RET ADR
                                                 ;NEEDS 1-6 CHAR FILENAME AT "MPGBGN"

003514' 004567  000452          DELETE:  JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003520' 004767  000236                   JSR     PC,ESUSAV               ;DO COMMON SETUP FUNCTIONS
003524' 012767  002400' 002632           MOV     #OFNAM,NFNAME           ;INIT FILENAME ADR
003532' 004775  177762                   JSR     PC,@DLT(R5)             ;GO TO DEVICE'S DELETE ROUT
003536' 000167  177314          EJNRET:  JMP     EXNRET                  ;EXIT TO MPG VIA NORMAL RETURN


                                         ;ZERO SAVE DEVICE'S DIRECTORY

                                                 ;JSR    R5,ZERO         S/R CALL
                                                 ;.WORD  ERRADR-.        REL ERR RET ADR

003542' 004567  000424          ZERO:    JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003546' 004767  001252                   JSR     PC,CLRBUF               ;INITIALIZE BUFFER AREA
003552' 004567  001222                   JSR     R5,BCLEAR               ;CLEAR OUTPUT DDB
003556' 002270'                          .WORD   ODHSU
003560' 000000                           .WORD   0
003562' 000122                           .WORD   DDBEND-ODHSU
003564' 004767  000172                   JSR     PC,ESUSAV               ;DO COMMON SETUP FUNCTIONS
003570' 004775  177760                   JSR     PC,@ZER(R5)             ;GO TO ZERO DIR ROUTINE
```

```
003574' 000760                          BR      EJNRET                  ;EXIT VIA NORMAL RETURN


                                        ;LIST FETCH DEV DIRECTORY

                                                ;JSR    R5,LIST         S/R CALL
                                                ;.WORD  ERRADR-.        REL ERR RET ADR

003576' 012567  174712          LIST:   MOV     (R5)+,LSTFLG            ;GET "ALL" & "FAST" FLAGS FROM MPG
003602' 004567  000364                  JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003606' 105267  174704                  INCB    PRTFLG                  ;SET PRINTER/TTY FLAG
003612' 004767  001730                  JSR     PC,CRLF                 ;ISSUE CR/LF
003616' 004767  000206                  JSR     PC,ESUFTH               ;DO COMMON SET UP
003622' 016703  174642                  MOV     TODAY,R3                ;GET TODAY'S DATE CODE
003626' 004567  001146                  JSR     R5,BCLEAR               ;SET FILENAME TO ALL ?'S
003632' 002256'                         .WORD   IFNAM
003634' 037477                          .ASCII  /??/
003636' 000011                          .WORD   9.
003640' 004767  002172                  JSR     PC,DATUPK               ;CONVERT & DISPLAY IT
003644' 004775  177756                  JSR     PC,@DRT(R5)             ;GO TO DIRECTORY LIST ROUT
003650' 004767  001672                  JSR     PC,CRLF                 ;ISSUE A CR/LF
003654' 000730                          BR      EJNRET                  ;EXIT VIA NORMAL RETURN


                                        ;BOOT FROM FETCH DEVICE

                                                ;JSR    R5,BOOT         S/R CALL
                                                ;.WORD  ERRADR-.        REL ERR RET ADR

003656' 004567  000310          BOOT:   JSR     R5,CREGSV               ;SAVE REG'S R0-R5
003662' 004767  000142                  JSR     PC,ESUFTH               ;DO COMMON SET UP
003666' 004775  177754                  JSR     PC,@XBT(R5)             ;DO PARTIAL BOOT
003672' 005065  000004                  CLR     XBA(R5)                 ;SET READ'S MEM ADR TO 0
003676' 010065  000006                  MOV     R0,XDT(R5)              ;STORE BOOT BLOCK #
003702' 004767  000610                  JSR     PC,BKRD0                ;DO READ OF THE BOOT BLK
003706' 016500  000012                  MOV     XRD(R5),R0              ;GET DEVICE'S READ CMND
003712' 042700  000001                  BIC     #1,R0                   ;RESET THE "GO" BIT
003716' 010075  000000                  MOV     R0,@(R5)                ;LOAD DEV CSR WITH UNIT #
003722' 011500                          MOV     (R5),R0                 ;STORE CSR ADR IN R0
003724' 005077  005210                  CLR     @CTRDRA                 ;RESET TTY INT ENABLE
003730' 032777  000001  000012'         BIT     #MMVER,@CSYSFW          ;RUNNING UNDER MEM MGMNT?
003736' 001410                          BEQ     10$                     ;Y,N-10$
003740' 005037  177572                  CLR     @#MMMR0                 ;DISABLE MEM MGMNT
003744' 032777  000040  000012'         BIT     #UNIMAP,@CSYSFW         ;USING THE UNIBUS MAP?
003752' 001402                          BEQ     10$                     ;Y,N-10$
003754' 005037  172516                  CLR     @#MMMR3                 ;DISABLE IT
003760' 005007                  10$:    CLR     PC                      ;BRANCH TO CODE JUST READ - (LOC 0)
```

```
1000                                            .SBTTL  SUBROUTINES FOR REQUIRED MPG FILE ROUTINES
1001
1002
1003                                            ;"SAVE" & OUTPUT DDB COMMON SETUP
1004
1005                                                    ;JSR    PC,ESETUP       S/R CALL
1006
1007  003762' 012767  051525  176416  ESUSAV:  MOV     #"US,OFNAM+6            ;SET ".USR" EXT IN OUTPUT DDB
1008  003770' 112767  000122  176412           MOVB    #'R,OFNAM+8.
1009  003776' 012767  040523  002636           MOV     #"SA,EFDEVN             ;SET FUNCT DEV NAME TO "SAVE"
1010  004004' 012767  042526  002632           MOV     #"VE,EFDEVN+2
1011  004012' 004767  000100                   JSR     PC,ESUOS                ;SET UP FOR SAVE DEV HNDLR
1012  004016' 016702  000006G                  MOV     CSAVEI+6,R2             ;GET SAVE'S UNIT #
1013  004022' 004767  000176                   JSR     PC,ESUOPT               ;INITIALIZE OUTPUT DDB
1014  004026' 000207                           RTS     PC                      ;EXIT IN-LINE
1015
1016
1017                                            ;"FETCH" & INPUT DDB COMMON SET UP
1018
1019                                                    ;JSR    PC,ESUFTH       S/R CALL
1020                                                    ;JSR    PC,ESUFT1       S/R CALL
1021                                                    ;JSR    PC,ESUFNM       S/R CALL
1022
1023  004030' 004767  000016           ESUFTH: JSR     PC,ESUFNM               ;SET UP "USR" EXT & FTCH I.D.
1024  004034' 004767  000044           ESUFT1: JSR     PC,ESUIF                ;SET UP FOR FETCH DEV HNDLR
1025  004040' 016702  000006G                  MOV     CFTCHI+6,R2             ;GET FETCH'S UNIT #
1026  004044' 004767  000146                   JSR     PC,ESUIPT               ;INITIALIZE INPUT DDB
1027  004050' 000207                           RTS     PC                      ;EXIT IN-LINE
1028
1029  004052' 012767  051525  176204  ESUFNM:  MOV     #"US,IFNAM+6            ;SET ".USR" EXT IN INPUT DDB
1030  004060' 112767  000122  176200           MOVB    #'R,IFNAM+8.
1031  004066' 012767  052106  002546           MOV     #"FT,EFDEVN             ;SET FUNCT DEV NAME TO "FTCH"
1032  004074' 012767  044103  002542           MOV     #"CH,EFDEVN+2
1033  004102' 000207                           RTS     PC                      ;EXIT IN-LINE
1034
1035
1036                                            ;SET UP DEVICE HANDLER ADR IN INPUT OR OUTPUT DDB
1037
1038                                                    ;JSR    PC,ESUIF        S/R CALL
1039                                                    ;JSR    PC,ESUOS        S/R CALL
1040
1041  004104' 012700  000004G          ESUIF:  MOV     #CFTCHI+4,R0            ;SET UP "FETCH" MDL CODE ADR
1042  004110' 012701  002146'                  MOV     #IDHSU,R1               ;SET UP INPUT DDB ADR
1043  004114' 000404                           BR      ESUCOM                  ;GO TO COMMON PROCESSING
1044
1045  004116' 012700  000004G          ESUOS:  MOV     #CSAVEI+4,R0            ;SET UP "SAVE" MDL CODE ADR
1046  004122' 012701  002270'                  MOV     #ODHSU,R1               ;SET UP OUTPUT DDB ADR
1047  004126' 012702  012006'          ESUCOM: MOV     #EMDLTB,R2              ;GET ADR OF MDL TBL
1048  004132' 021022                   20$:    CMP     (R0),(R2)+              ;THIS THE DEV'S MDL?
1049  004134' 001404                           BEQ     30$                     ;N,Y-30$
1050  004136' 005722                           TST     (R2)+                   ;POINT TO NXT ENTRY
1051  004140' 005712                           TST     (R2)                    ;END OF THE TBL?
1052  004142' 001412                           BEQ     40$                     ;N,Y-40$ -- GO TO EXIT
1053  004144' 000772                           BR      20$                     ;GO CK NXT ENTRY
1054  004146' 011211                   30$:    MOV     (R2),(R1)               ;STORE DEVICE'S SET UP ADR
1055  004150' 020227  012010'                  CMP     R2,#PTSUAD              ;IS THIS PAPER TAPE?
```

```
1056  004154' 001005                    BNE     40$                     ;Y,N-40$
1057  004156' 020027  000004G           CMP     R0,#CSAVEI+4            ;IS THIS FOR THE SAVE DEV (PUNCH)?
1058  004162' 001002                    BNE     40$                     ;Y,N-40$
1059  004164' 062711  000010            ADD     #8.,(R1)                ;POINT ADR AT PT PUNCH SET UP  ADR
1060  004170' 000207          40$:      RTS     PC                      ;EXIT IN-LINE
1061
1062
1063                                    ;STORE REGISTERS & SAVE STACK POINTER
1064
1065                                            ;JSR    R5,CREGSV       S/R CALL
1066
1067  004172' 010446          CREGSV: MOV     R4,-(SP)                ;SAVE REG'S R4 THRU R0
1068  004174' 010346                  MOV     R3,-(SP)                ;R5 SAVED BY JSR TO THIS S/R
1069  004176' 010246                  MOV     R2,-(SP)
1070  004200' 010146                  MOV     R1,-(SP)
1071  004202' 010046                  MOV     R0,-(SP)
1072  004204' 010667  174302          MOV     SP,STKADR               ;SAVE STACK POINTER
1073  004210' 105067  174302          CLRB    PRTFLG                  ;RESET PRINTER/TTY FLAG
1074  004214' 010507                  MOV     R5,PC                   ;EXIT IN-LINE
```

```
1076                                            .SBTTL  FILE INITIALIZATION S/R'S REQUIRED FOR LOAD MEDIA HANDLERS
1077
1078
1079                                            ;INITIALIZE INPUT OR OUTPUT DDB S/R
1080
1081                                                    ;JSR    PC,ESUIPT       S/R CALL
1082                                                    ;R2 CONTAINS UNIT # UPON ENTRY
1083
1084  004216' 012705  002232'       ESUIPT: MOV     #INDEV,R5               ;SET UP INPUT DDB ADR
1085  004222' 000402                        BR      ESUIOC                  ;GO TO COMMON POINT
1086  004224' 012705  002354'       ESUOPT: MOV     #OUTDEV,R5              ;SET UP OUTPUT DDB ADR
1087  004230' 004775  177714        ESUIOC: JSR     PC,@XDHSU(R5)           ;GO TO HANDLER'S SET UP ADR
1088  004234' 010067  000020                MOV     R0,12$                  ;SET DEV PARAM INFO ADR
1089  004240' 010567  000012                MOV     R5,10$                  ;SET DDB ADR
1090  004244' 062767  177754  000004       ADD     #XBT,10$
1091  004252' 004567  000466                JSR     R5,BMOVE                ;MOVE DEV INFO TO DDB
1092  004256' 000000                10$:    .WORD   XXXX                    ;DEST
1093  004260' 000000                12$:    .WORD   XXXX                    ;SRC
1094  004262' 000050                        .WORD   40.                     ;CNT
1095  004264' 010504                        MOV     R5,R4                   ;MOVE ASCII FILENAME TO DDB
1096  004266' 062704  000024                ADD     #XXNAM,R4
1097  004272' 016703  174174                MOV     MPGST,R3
1098  004276' 012324                        MOV     (R3)+,(R4)+
1099  004300' 012324                        MOV     (R3)+,(R4)+
1100  004302' 011314                        MOV     (R3),(R4)
1101  004304' 005065  177722                CLR     XFLCNT(R5)              ;RESET FILES PROCESSED CNT
1102  004310' 000207                        RTS     PC                      ;EXIT IN-LINE
1103
1104
1105                                            ;INITIALIZE OUTPUT FOR LINKED FILE
1106
1107                                                    ;JSR    PC,EINTOL       S/R CALL
1108
1109  004312' 012705  002354'       EINTOL: MOV     #OUTDEV,R5              ;SET UP OUTPUT DDB ADR
1110  004316' 005065  177716                CLR     XWCTR(R5)               ;RESET BLOCKS WRITTEN CNT
1111  004322' 016767  174142  175764        MOV     TODAY,OSVDAT            ;SET TODAY'S DATE IN DDB
1112  004330' 005067  175766                CLR     OBKLGT                  ;CLEAR FILE LENGTH
1113  004334' 004775  177766                JSR     PC,@ETR(R5)             ;GO TO DEV'S "ENTER" ROUT
1114  004340' 012767  177777  176024        MOV     #-1,OUBKCT              ;SET UP LINKED FILE CODE
1115  004346' 004775  177772                JSR     PC,@ALC(R5)             ;ALLOCATE THE FIRST BLOCK
1116  004352' 005265  177716                INC     XWCTR(R5)               ;INCR BLOCKS WRITTEN CNT
1117  004356' 016767  175742  175734        MOV     OLSTBK,O1STBK           ;STORE FIRST BLOCK #
1118  004364' 005267  175732                INC     OBKLGT                  ;INCR # OF BLKS USED CNT
1119  004370' 012767  000001  174546        MOV     #1,BUF                  ;INITIALIZE BLOCK # IN BUF
1120  004376' 000207                        RTS     PC                      ;EXIT IN-LINE
```

```
                                        .SBTTL  FILE READ S/R'S FOR LOAD MEDIA HANDLERS


                                        ;READ LINKED FILE BLOCK INTO BUF

                                                ;JSR    PC,ERDLNK       S/R CALL
                                                ;BR     ERRADR          EXECUTED IF AN ERROR

004400' 012705  002232'         ERDLNK: MOV     #INDEV,R5               ;GET INPUT DDB ADR
004404' 112767  000001  174110          MOVB    #1,PIPFLG               ;SET LINK MODE FLAG
004412' 005765  000006                  TST     XDT(R5)                 ;IS THERE ANOTHER BLOCK?
004416' 001410                          BEQ     ERDERR                  ;Y,N-ERDERR
004420' 004767  000060                  JSR     PC,BKREAD               ;GO READ THE BLOCK
004424' 016765  174514  000006          MOV     BUF,XDT(R5)             ;SAVE NEXT BLOCK ADDRESS
004432' 062716  000002          ENMRET: ADD     #2,(SP)                 ;SET RET ADR FOR NORMAL +2 POINT
004436' 000207                  ERET:   RTS     PC                      ;EXIT IN-LINE

004440' 012705  006626'         ERDERR: MOV     #EOFMSG,R5              ;SET UP END OF FILE MSG ADR
004444' 004767  002062                  JSR     PC,EXERPT               ;ISSUE EOF ERROR MSG
004450' 000565                          BR      CLRBUF                  ;CLEAR THE BUFFER & EXIT


                                        ;INPUT OR OUTPUT NEXT BLOCK IN FILE

                                                ;JSR    PC,NXTBLK       S/R CALL

004452' 012765  000400  000002  NXTBLK: MOV     #256.,XWC(R5)           ;SET UP WORD COUNT
004460' 012765  001144'  000004         MOV     #BUF,XBA(R5)            ;SET UP DATA ADR
004466' 016765  174452  000006          MOV     BUF,XDT(R5)             ;GET NEXT BLOCK #
004474' 001760                          BEQ     ERET                    ;THIS LAST BLOCK? (N,Y-ERET)
004476' 062716  000002                  ADD     #2,(SP)                 ;SET UP FOR NORMAL RETURN
004502' 000471                          BR      EXYBK                   ;GO DO READ OR WRITE


                                        ;READ A BLOCK INTO THE BUFFER

                                                ;JSR    PC,BKREAD       S/R CALL
                                                ;JSR    PC,READBK       S/R CALL

004504' 004767  000314          BKREAD: JSR     PC,CLRBUF               ;CLEAR THE BUFFER
004510' 012765  001144'  000004         MOV     #BUF,XBA(R5)            ;SET UP READ DATA ADR
004516' 012765  000400  000002  BKRDO:  MOV     #256.,XWC(R5)           ;SET UP WORD COUNT
004524' 016565  000012  000010  READBK: MOV     XRD(R5),XCO(R5)         ;SET UP THIS DEV'S READ CMND
004532' 000455                          BR      EXYBK                   ;GO DO READ OR WRITE
```

```
                                        .SBTTL  FILE WRITE S/R'S FOR LOAD MEDIA HANDLERS


                                        ;WRITE A LINKED  FILE BLOCK

                                                ;JSR    PC,EWRITL       S/R CALL

004534' 004467  000156          EWRITL: JSR     R4,SAV04                ;SAVE WORK REGISTERS
004540' 012705  002354'                 MOV     #OUTDEV,R5              ;GET OUTPUT DOB ADR
004544' 112767  000001  173750          MOVB    #1,PIPFLG               ;SET LINK MODE FLAG
004552' 016767  175546  175602          MOV     OLSTBK,OUTDT            ;SET UP BLOCK NUMBER
004560' 005767  174360                  TST     BUF                     ;THIS LAST BLK TO BE WRITTEN?
004564' 001415                          BEQ     10$                     ;N,Y-10$
004566' 012767  177777  175576          MOV     #-1,OUBKCT              ;SET ALLOCATE LINKED FILE CODE
004574' 004775  177772                  JSR     PC,@ALC(R5)             ;GO HAVE DEV ALLOCATE NXT BLK
004600' 005267  175516                  INC     OBKLGT                  ;INCR REQUESTED BLOCK COUNT
004604' 016767  175514  174332          MOV     OLSTBK,BUF              ;LINK CURR BLOCK TO NEXT
004612' 004767  000014                  JSR     PC,WRTLC                ;GO WRITE  LINKED BLOCK & CLEAR BUF
004616' 000427                          BR      BKWRT1                  ;GO TO EXIT
004620' 004767  000006          10$:    JSR     PC,WRTLC                ;GO WRITE LINKED BLOCK & CLEAR BUF
004624' 004767  000054                  JSR     PC,FCLOSE               ;GO CLOSE THE FILE
004630' 000422                          BR      BKWRT1                  ;GO TO EXIT


                                        ;WRITE 256 WORD BLK, INCR BLK CNT, & CLEAR BUFFER

                                                ;JSR    PC,WRTLC        S/R  CALL

004632' 004767  000006          WRTLC:  JSR     PC,BLKWRT               ;GO WRITE THE BLOCK
004636' 005265  177716                  INC     XWCTR(R5)               ;INCR BLKS WRITTEN CNT
004642' 000470                          BR      CLRBUF                  ;GO CLEAR BUF & EXIT


                                        ;WRITE 256 WORD BLOCK FROM BUF (BLK # ALREADY SET)

                                                ;JSR    PC,BLKWRT       S/R CALL
                                                ;JSR    PC,WRTBLK       S/R CALL

004644' 012765  001144' 000004  BLKWRT: MOV     #BUF,XBA(R5)            ;SET UP DATA ADR
004652' 012765  000400  000002  BKWRTO: MOV     #256.,XWC(R5)           ;SET UP WORD COUNT
004660'                         WRTDIR:
004660' 016565  000014  000010  WRTBLK: MOV     XWT(R5),XCO(R5)         ;SET UP THIS DEV'S WRITE CMND
004666' 004467  000024          EXYBK:  JSR     R4,SAV04                ;SAVE REGISTERS
004672' 004775  177774                  JSR     PC,@XSV(R5)             ;GO READ OR WRITE THIS BLOCK
004676' 004767  000026          BKWRT1: JSR     PC,RST04                ;RESTORE REG'S 0-4
004702' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;CLOSE OUTPUT FILE

                                                ;JSR    PC,FCLOSE       S/R CALL

004704' 004775  177764          FCLOSE: JSR     PC,@CLS(R5)             ;GO TO DEVICE'S CLOSE ROUT
004710' 005065  177716                  CLR     XWCTR(R5)               ;CLEAR "OUTPUT FILE OPEN" FLAG
004714' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                .SBTTL  DATA S/R'S REQUIRED FOR LOAD MEDIA HANDLERS


                        ;***************
                        ;
                        ;  SAV04:  SAVE REGISTERS R0 THRU R4 ON THE STACK
                        ;
                        ;       LINK:   JSR     R4,SAV04        S/R CALL
                        ;
                        ;***************

004716' 010346          SAV04:  MOV     R3,-(SP)                ;SAVE R3 THRU R0
004720' 010246                  MOV     R2,-(SP)
004722' 010146                  MOV     R1,-(SP)
004724' 010046                  MOV     R0,-(SP)
004726' 010407                  MOV     R4,PC                   ;EXIT -- R4 IS ALREADY SAVED


                        ;***************
                        ;
                        ;  RST04:  RESTORE REGISTERS R0 THRU R4 SAVED BY SAV04 S/R
                        ;
                        ;       LINK:   JSR     PC,RST04        S/R CALL
                        ;
                        ;***************

004730' 012604          RST04:  MOV     (SP)+,R4                ;SET UP RETURN ADR
004732' 012600                  MOV     (SP)+,R0                ;RELOAD R0 THRU R3
004734' 012601                  MOV     (SP)+,R1
004736' 012602                  MOV     (SP)+,R2
004740' 012603                  MOV     (SP)+,R3
004742' 000204                  RTS     R4                      ;EXIT IN-LINE & RELOAD R4


                        ;***************
                        ;
                        ;  BMOVE:  MOVE BYTE FIELDS S/R
                        ;
                        ;       LINK:   JSR     R5,BMOVE        S/R CALL
                        ;               .WORD   DESTADR         DESTINATION ADR (ABSOLUTE)
                        ;               .WORD   SRCADR          SOURCE ADR (ABSOLUTE)
                        ;               .WORD   COUNT           NUMBER OF BYTES TO MOVE
                        ;
                        ;***************

004744' 012767  112120  000012  BMOVE:  MOV     #112120,EMINST          ;SET UP "MOVB (R1)+,(R0)+" INST
004752' 004467  177740          EMCOM:  JSR     R4,SAV04                ;SAVE REGISTERS
004756' 012500                  MOV     (R5)+,R0                ;GET DESTINATION ADR
004760' 012501                  MOV     (R5)+,R1                ;GET SOURCE ADR OR CLEAR DATA
004762' 012502                  MOV     (R5)+,R2                ;GET BYTE COUNT
004764' 000000          EMINST: .WORD   XXXX                    ;"MOVB (R1)+,(R0)+" OR
                                                                ;"MOVB R1,(R0)+" INST
004766' 005302                  DEC     R2                      ;DECR BYTE COUNT
004770' 001375                  BNE     EMINST                  ;CNT = 0? (Y,N-EMINST)
004772' 004767  177732          ECOMEX: JSR     PC,RST04                ;RESTORE SAVED REGISTERS
004776' 000205                  RTS     R5                      ;EXIT IN-LINE
```

```
1280                                    ;****************
1281                                    ;
1282                                    ;  BCLEAR:  CLEAR BYTE FIELDS TO A SPECIFIC VALUE S/R
1283                                    ;
1284                                    ;       LINK:   JSR     R5,BCLEAR       S/R CALL
1285                                    ;               .WORD   DESTADR         DESTINATION ADR (ABSOLUTE)
1286                                    ;               .WORD   DATA            DATA BYTE TO CLEAR WITH
1287                                    ;               .WORD   COUNT           NUMBER OF BYTES TO CLEAR
1288                                    ;
1289                                    ;       USES "BMOVE" S/R
1290                                    ;
1291                                    ;****************
1292
1293  005000' 012767  110120  177756  BCLEAR: MOV     #110120,EMINST          ;SET UP "MOVB R1,(R0)+" INST
1294  005006' 000761                          BR      EMCOM                   ;GO TO "BMOVE" S/R
1295
1296
1297                                    ;****************
1298                                    ;
1299                                    ;  CRPMAP:  CLEAR 128 BYTES TO 0'S STARTING AT "MAPBUF"
1300                                    ;
1301                                    ;       LINK:   JSR     PC,CRPMAP       S/R CALL
1302                                    ;
1303                                    ;****************
1304
1305  005010' 004567  177764          CRPMAP: JSR     R5,BCLEAR               ;GO CLEAR THE MAPBUF
1306  005014' 000544'                         .WORD   MAPBUF
1307  005016' 000000                          .WORD   0
1308  005020' 000200                          .WORD   128.
1309  005022' 000207                          RTS     PC                      ;EXIT IN-LINE
1310
1311
1312                                    ;****************
1313                                    ;
1314                                    ;  CLRBUF:  CLEAR 510 BYTES TO 0'S STARTING AT "BUF+2"
1315                                    ;
1316                                    ;       LINK:   JSR     PC,CLRBUF       S/R CALL
1317                                    ;
1318                                    ;****************
1319
1320  005024' 004567  177750          CLRBUF: JSR     R5,BCLEAR               ;GO CLEAR THE BUFFER
1321  005030' 001146'                         .WORD   BUF+2
1322  005032' 000000                          .WORD   0
1323  005034' 000776                          .WORD   510.
1324  005036' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                ;****************
                                ;
                                ;  CMPNAM:  COMPARE TWO 9 BYTE FIELDS
                                ;
                                ;       LINK:   JSR     R5,CMPNAM       S/R CALL
                                ;               BR      XXXX            BR IF UNEQUAL
                                ;
                                ;       IF DATA IS UNEQUAL, RETURN WILL BE SO THAT THE BR
                                ;       INST IS EXECUTED.  IF EQUAL, RETURN WILL BE IN-LINE AFTER
                                ;       THE BR INST.
                                ;
                                ;****************
                                
005040' 004467  177652  CMPNAM: JSR     R4,SAV04        ;SAVE WORK REGISTERS
005044' 012700  000024          MOV     #XXNAM,R0       ;SET UP DDB DISPLACEMENT
005050' 066600  000012          ADD     12(SP),R0       ;ADD IN CURR DDB ADR
005054' 012701  000524'         MOV     #TXNAM,R1       ;SET UP SECOND FILENAME ADR
005060' 012702  000011          MOV     #9.,R2          ;SET CNT TO 9
005064' 122021          10$:    CMPB    (R0)+,(R1)+     ;DATA CHAR MATCH?
005066' 001011          BNE     30$             ;Y,N-30$
005070' 005302          20$:    DEC     R2              ;DECR CHAR CNT
005072' 001374          BNE     10$             ;CNT = 0? (Y,N-10$)
005074' 004567  177644          JSR     R5,BMOVE        ;SAVE FILE'S NAME IN CASE THERE
005100' 000536'                 .WORD   REALNM          ;WERE ?'S IN DDB'S FILE NAME
005102' 000524'                 .WORD   TXNAM
005104' 000006                  .WORD   6.
005106' 005725                  TST     (R5)+           ;INCR PAST BR INST
005110' 000730                  BR      ECOMEX          ;GO TO EXIT AFTER BR
005112' 126027  177777  000077  30$:    CMPB    -1(R0),#'?      ;THIS A "DON'T CARE" CHAR?
005120' 001324          BNE     ECOMEX          ;Y,N-ECOMEX
005122' 000762          BR      20$             ;GO CK NEXT CHAR

                                ;****************
                                ;
                                ;  DELAY:  THIS S/R PERFORMS A COMPUTE BOUND DELAY
                                ;
                                ;       LINK:   JSR     PC,DELAY        S/R CALL
                                ;
                                ;****************

005124' 005046          DELAY:  CLR     -(SP)           ;INITIALIZE LOOP COUNT
005126' 005316          40$:    DEC     (SP)            ;DECR LOOP CNT
005130' 100776          BMI     40$             ;DONE LOOP 32K TIMES? (Y,N-40$)
005132' 005726          TST     (SP)+           ;CLEAR CNTR OFF STACK
005134' 000207          RTS     PC              ;EXIT IN-LINE
```

```
                                .SBTTL  RADIX 50 CONVERSION S/R'S FOR LOAD MEDIA HANDLERS


                        ;***************
                        ;
                        ;  PAKNAM:      CONVERT 9 CHARACTER ASCII FILE NAME TO
                        ;               3 RADIX 50 WORDS S/R
                        ;
                        ;       LINK:   JSR     R5,PAKNAM       S/R CALL
                        ;               .WORD   ASCIADR         ADR OF 9 ASCII CHAR'S
                        ;               .WORD   R50ADR          ADR OF 3 OUTPUT RAD50 WORDS
                        ;
                        ;***************
005136' 004467  177554  PAKNAM: JSR     R4,SAV04                ;SAVE WORK REG'S
005142' 012501                  MOV     (R5)+,R1                ;GET ADR OF ASCII DATA
005144' 012500                  MOV     (R5)+,R0                ;GET ADR OF RAD50 WORDS
005146' 004567  000012          JSR     R5,PACK                 ;PACK FIRST 6 ASCII INTO 2 RAD50 WORDS
005152' 062700  000004          ADD     #4,R0                   ;INCR TO EXT STORAGE
005156' 004567  000012          JSR     R5,PACK1                ;PACK LAST 3 ASCII INTO 1 WORD
005162' 000456                  BR      EPKEX                   ;GO TO EXIT


                        ;***************
                        ;
                        ;  PACK:        DEPENDING UPON ENTRY POINT, EITHER CONVERTS
                        ;  PACK1:       6 OR 3 ASCII WORDS TO EITHER 2 OR 1
                        ;               RAD50 WORDS. REG R1 CONTAINS ADR OF ASCII
                        ;               DATA UPON ENTRY AND POINTS TO THE NEXT LOCATION
                        ;               AFTER THE ASCII DATA UPON EXIT. REG R0 CONTAINS
                        ;               ADR OF OUTPUT RAD50 WORDS UPON ENTRY.
                        ;
                        ;       LINK:   JSR     R5,PACK         6/2 S/R CALL
                        ;                       ,PACK1          3/1 S/R CALL
                        ;
                        ;***************
005164' 012767  177776  000006  PACK:   MOV     #-2,EPKCNT              ;SET OUTPUT WORD CNT TO 2
005172' 000403                  BR      EPKCOM                  ;GO TO COMMON POINT

005174' 012727  177777          PACK1:  MOV     #-1,(PC)+               ;SET OUTPUT WORD CNT TO 1
005200' 000000                  EPKCNT: .WORD   0                       ;S/R LOOP COUNT
005202' 004467  177510          EPKCOM: JSR     R4,SAV04                ;STORE WORK REG'S
005206' 012704  177775          10$:    MOV     #-3,R4                  ;INITIALIZE INTERNAL LOOP CNT
005212' 005002                  CLR     R2                      ;CLEAR SUM
005214' 112103                  20$:    MOVB    (R1)+,R3                ;GET NEXT ASCII CHAR
005216' 122703  000040          CMPB    #40,R3                  ;CHAR A BLANK?
005222' 001410                  BEQ     30$                     ;N,Y-30$
005224' 122703  000044          CMPB    #'$,R3                  ;A DOLLAR SIGN?
005230' 001411                  BEQ     50$                     ;N,Y-50$
005232' 120327  000101          CMPB    R3,#'A                  ;LESS THAN AN "A"?
005236' 103404                  BLO     40$                     ;N,Y-40$
005240' 162703  000040          SUB     #40,R3                  ;ADJUST FOR A-Z CHAR
005244' 162703  000016  30$:    SUB     #16,R3                  ;ADJUST FURTHER
005250' 162703  000011  40$:    SUB     #11,R3
005254' 162703  000011  50$:    SUB     #11,R3
```

```
1429  005260' 006302                        ASL     R2                      ;MULTIPLY CURRENT CHAR TOTAL
1430  005262' 006302                        ASL     R2                      ;BY DECIMAL 40
1431  005264' 006302                        ASL     R2
1432  005266' 010246                        MOV     R2,-(SP)
1433  005270' 006302                        ASL     R2
1434  005272' 006302                        ASL     R2
1435  005274' 062602                        ADD     (SP)+,R2
1436  005276' 060302                        ADD     R3,R2                   ;INCLUDE CURR CHAR
1437  005300' 005204                        INC     R4                      ;DONE 3 CHAR'S?
1438  005302' 002744                        BLT     20$                     ;Y,N-20$
1439  005304' 010220                        MOV     R2,(R0)+                ;STORE RAD50 WORD
1440  005306' 005267  177666                INC     EPKCNT                  ;DONE # OF WORDS SPECIFIED?
1441  005312' 002735                        BLT     10$                     ;Y,N-10$
1442  005314' 010166  000002       EPKPEX: MOV     R1,2(SP)                ;REPLACE R1 ON STK WITH FINAL ASCII ADR
1443  005320' 004767  177404       EPKEX:  JSR     PC,RST04                ;RESTORE REG'S
1444  005324' 000205                        RTS     R5                      ;EXIT IN-LINE
```

```
1446                                    ;***************
1447                                    ;
1448                                    ;  UPKNAM:      UNPACK 3 RAD50 WORDS TO 9 ASCII CHARACTERS
1449                                    ;                WHICH ARE A FILENAME.
1450                                    ;
1451                                    ;       LINK:   JSR     R5,UPKNAM       S/R CALL
1452                                    ;               .WORD   ASCIADR         ADR OF 9 ASCII CHAR'S OF OUTPUT
1453                                    ;               .WORD   R50ADR          ADR OF 3 INPUT RAD50 WORDS
1454                                    ;
1455                                    ;***************
1456
1457  005326' 004467  177364    UPKNAM: JSR     R4,SAV04                ;SAVE WORK REG'S
1458  005332' 012501                    MOV     (R5)+,R1                ;GET ASCII OUTPUT DATA ADR
1459  005334' 012500                    MOV     (R5)+,R0                ;GET INPUT RAD50 WORDS ADR
1460  005336' 004567  000012            JSR     R5,UNPACK               ;GO UNPACK FIRST TWO WORDS
1461  005342' 062700  000004            ADD     #4,R0                   ;ADJUST INPUT DATA ADR
1462  005346' 004567  000012            JSR     R5,UPACK1               ;GO UNPACK LAST WORD
1463  005352' 000762                    BR      EPKEX                   ;GO TO EXIT
1464
1465
1466                                    ;***************
1467                                    ;
1468                                    ;  UNPACK:      DEPENDING UPON ENTRY POINT, EITHER CONVERTS
1469                                    ;  UPACK1:      2 OR 1 RAD50 WORDS TO EITHER 6 OR 3
1470                                    ;               ASCII CHARACTERS. REG R1 CONTAINS ADR OF ASCII
1471                                    ;               OUTPUT DATA UPON ENTRY AND POINTS TO THE NEXT
1472                                    ;               LOCATION AFTER THE ASCII DATA UPON EXIT. REG R0
1473                                    ;               CONTAINS THE ADR OF THE INPUT RAD50 WORDS UPON ENTRY.
1474                                    ;
1475                                    ;       LINK:   JSR     R5,UNPACK       2/6 S/R CALL
1476                                    ;                       ,UPACK1         1/3 S/R CALL
1477                                    ;
1478                                    ;***************
1479
1480  005354' 012767  177776  177616 UNPACK: MOV     #-2,EPKCNT              ;SET INPUT WORD CNT TO 2
1481  005362' 000403                    BR      EUPKCM                  ;GO TO COMMON POINT
1482
1483  005364' 012767  177777  177606 UPACK1: MOV     #-1,EPKCNT              ;SET INPUT WORD CNT TO 1
1484  005372' 004467  177320         EUPKCM: JSR     R4,SAV04                ;SAVE WORK REG'S
1485  005376' 012704  177775         5$:     MOV     #-3,R4                  ;INITIALIZE INTERNAL LOOP CNT
1486  005402' 011000                          MOV     (R0),R0                 ;GET CURR RAD50 WORD
1487  005404' 012702  005504'                 MOV     #ECOEFF,R2              ;GET COEFFICIENT TBL ADR
1488  005410' 005003                  10$:    CLR     R3                      ;CLEAR QUOTIENT REG
1489  005412' 020012                  20$:    CMP     R0,(R2)                 ;DONE WITH PSEUDO DIVIDE?
1490  005414' 103403                          BLO     30$                     ;N,Y-30$
1491  005416' 161200                          SUB     (R2),R0                 ;SUBT CURR COEFF VALUE
1492  005420' 005203                          INC     R3                      ;INCR QUOTIENT
1493  005422' 000773                          BR      20$                     ;LOOP AGAIN
1494  005424' 105703                  30$:    TSTB    R3                      ;QUOTIENT ZERO? (A BLANK)
1495  005426' 001406                          BEQ     40$                     ;N,Y-40$
1496  005430' 120327  000033                  CMPB    R3,#33                  ;QUOT A "$"?
1497  005434' 001407                          BEQ     60$                     ;N,Y-60$
1498  005436' 003004                          BGT     50$                     ;"." OR "0-9"? (N,Y-50$)
1499  005440' 062703  000040                  ADD     #40,R3                  ;MUST BE "A-Z"; ADJUST ACCORDINGLY
1500  005444' 062703  000016          40$:    ADD     #16,R3                  ;ADJUST VALUE FURTHER
1501  005450' 062703  000011          50$:    ADD     #11,R3
```

```
1502  005454' 062703 000011        60$:    ADD     #11,R3
1503  005460' 110321                       MOVB    R3,(R1)+        ;STORE OUTPUT ASCII CHAR
1504  005462' 005722                       TST     (R2)+           ;POINT TO NXT COEFF
1505  005464' 005204                       INC     R4              ;DONE 3 OUTPUT CHAR'S?
1506  005466' 002750                       BLT     10$             ;Y,N-10$
1507  005470' 011600                       MOV     (SP),R0         ;RESTORE ORG RAD50 WORD ADR
1508  005472' 005720                       TST     (R0)+           ;POINT IT AT NEXT WORD
1509  005474' 005267 177500                INC     EPKCNT          ;DONE # OF WORDS SPECIFIED?
1510  005500' 002736                       BLT     5$              ;Y,N-5$
1511  005502' 000704                       BR      EPKPEX          ;GO TO EXIT
1512
1513  005504' 003100               ECOEFF: .WORD   1600.
1514  005506' 000050                       .WORD   40.
1515  005510' 000001                       .WORD   1
```

```
                                        .SBTTL  TYPE S/R'S REQUIRED FOR LOAD MEDIA HANDLERS


                                ;***************
                                ;
                                ;  CHROUT:  TYPE THE ASCII CHAR IN REG R2 S/R
                                ;
                                ;       LINK:   JSR     PC,CHROUT       S/R CALL
                                ;
                                ;***************

005512' 110267  000016          CHROUT: MOVB    R2,ETYPWK               ;STORE CHAR AT COMMON LOC
005516' 004467  177174          ECOMTY: JSR     R4,SAV04                ;SAVE WORK REGISTERS
005522' 004567  001512                  JSR     R5,CLIST                ;GO DO THE WRITE OF ONE CHAR
005526' 000006                          .WORD   ETYPWK-.
005530' 000001                          .WORD   1
005532' 000536                          BR      ETYEX                   ;GO TO EXIT

005534' 000005                  ETYPWK: .BLKW   5                       ;TYPE WORK/CHAR AREA

                                ;***************
                                ;
                                ;  CRLF:  THESE TWO S/R'S ARE USED TO TYPE A CARRIAGE
                                ;  TAB:   RETURN/LINE FEED AND TAB, RESPECTIVELY
                                ;
                                ;       LINK:   JSR     PC,CRLF         S/R CALL
                                ;       LINK:   JSR     PC,TAB          S/R CALL
                                ;
                                ;***************

005546' 004467  177144          CRLF:   JSR     R4,SAV04                ;SAVE REGISTERS
005552' 004567  001462                  JSR     R5,CLIST                ;ISSUE CR/LF CHARACTERS
005556' 001026                          .WORD   EXERM-.
005560' 000002                          .WORD   2
005562' 000522                          BR      ETYEX                   ;GO TO EXIT

005564' 112767  000011  177742  TAB:    MOVB    #011,ETYPWK             ;SET UP TAB CHAR
005572' 000751                          BR      ECOMTY                  ;GO TYPE IT

                                ;***************
                                ;
                                ;  $TYPNM:  $TYPNM WILL TYPE A CR/LF FOLLOWED BY THE NINE
                                ;  $TPNM1:  CHARACTER FILENAME STARTING AT THE ADDRESS STORED AT
                                ;           LOCATION "$TXNAM".  IN THE DISPLAYED DATA A PERIOD WILL BE
                                ;           INSERTED BETWEEN THE FILENAME AND ITS EXTENSION.
                                ;           $TPNM1 DOES THE SAME FUNCTION BUT WITHOUT THE
                                ;           LEADING CR/LF.
                                ;
                                ;       LINK:   JSR     PC,$TYPNAM      S/R CALL
                                ;                         ,$TYPNM1
                                ;
                                ;***************

005574' 004767  177746          $TYPNM: JSR     PC,CRLF                 ;ISSUE CR/LF
005600' 004467  177112          $TPNM1: JSR     R4,SAV04                ;SAVE WORK REGISTERS
005604' 016700  172700                  MOV     $TXNAM,R0               ;GET FILENAME ADR
```

```
1573  005610' 012701 005534'          MOV     #ETYPWK,R1              ;GET CHAR WORK AREA ADR
1574  005614' 012702 001406          MOV     #1406,R2                ;SET UP TWO LOOP COUNTS
1575  005620' 112021          10$:    MOVB    (R0)+,(R1)+             ;MOVE CHAR OF FILENAME
1576  005622' 105302                  DECB    R2                      ;DONE 6 CHAR FILENAME?
1577  005624' 001375                  BNE     10$                     ;Y,N-10$
1578  005626' 112721 000056           MOVB    #'.,(R1)+               ;PUT PERIOD IN TYPE DATA
1579  005632' 000302                  SWAB    R2                      ;GET SECOND LOOP COUNT
1580  005634' 112021          20$:    MOVB    (R0)+,(R1)+             ;MOVE CHAR OF EXT
1581  005636' 105302                  DECB    R2                      ;DONE 3 CHAR EXT?
1582  005640' 001375                  BNE     20$                     ;Y,N-20$
1583  005642' 004567 001372           JSR     R5,CLIST                ;TYPE THE FILENAME & EXT
1584  005646' 177666                  .WORD   ETYPWK-.
1585  005650' 000012                  .WORD   10.
1586  005652' 000466                  BR      ETYEX                   ;GO TO S/R EXIT
1587
1588
1589                          ;***************
1590                          ;
1591                          ;  MES:  THIS S/R WILL TYPE THE ASCII DATA STARTING AT THE
1592                          ;        SUPPLIED ADDRESS AND UP TO BUT NOT INCLUDING
1593                          ;        THE FIRST 00 CHARACTER.
1594                          ;
1595                          ;     LINK:  JSR     R5,MES          S/R CALL
1596                          ;            .WORD   DATADR          DATA ADR (ABSOLUTE)
1597                          ;
1598                          ;***************
1599
1600  005654' 004467 177036   MES:    JSR     R4,SAV04                ;SAVE WORK REG'S
1601  005660' 012500                  MOV     (R5)+,R0                ;GET DATA ADR
1602  005662' 010001                  MOV     R0,R1                   ;COMPUTE CHAR CNT
1603  005664' 105721          40$:    TSTB    (R1)+
1604  005666' 001376                  BNE     40$
1605  005670' 160001                  SUB     R0,R1
1606  005672' 005301                  DEC     R1
1607  005674' 005701                  TST     R1                      ;CHAR CNT = 0?
1608  005676' 001411                  BEQ     EMESEX                  ;N,Y-EMESEX
1609  005700' 012702 005716'          MOV     #44$,R2                 ;GET S/R INFO STORAGE ADR
1610  005704' 160200                  SUB     R2,R0                   ;MAKE DATA ADR RELATIVE
1611  005706' 010022                  MOV     R0,(R2)+                ;STORE DATA ADR
1612  005710' 010112                  MOV     R1,(R2)                 ;STORE CHAR CNT
1613  005712' 004567 001322           JSR     R5,CLIST                ;TYPE THE MSG
1614  005716' 000000          44$:    .WORD   XXXX
1615  005720' 000000                  .WORD   XXXX
1616  005722' 004767 177002   EMESEX: JSR     PC,RST04                ;RESTORE WORK REG'S
1617  005726' 000205                  RTS     R5                      ;EXIT IN-LINE
```

```
                                ;****************
                                ;
                                ;  BCDCV:  CONVERTS THE BINARY NUMBER IN R3 TO THE NUMBER OF
                                ;          DECIMAL ASCII DIGITS SPECIFIED BY THE COUNT WHICH FOLLOWS
                                ;          THE JSR. ALSO TYPE THESE DIGITS.
                                ;
                                ;       LINK:   JSR     R5,@BCDCV       S/R CALL
                                ;               .WORD   DIGITCNT        # OF DIGITS TO PRINT
                                ;
                                ;       R3 CONTAINS BINARY # TO CONVERT
                                ;
                                ;****************

005730' 004467 176762   BCDCV:  JSR     R4,SAV04                ;SAVE WORK REGISTERS
005734' 010300                  MOV     R3,R0                   ;SET WORD UP FOR S/R
005736' 004577 000010'          JSR     R5,@BTASLB              ;CONVERT # TO DECIMAL ASCII
005742' 000034                  .WORD   EBCDWA-.
005744' 012500                  MOV     (R5)+,R0                ;GET # OF DIGITS WANTED
005746' 012701 000006           MOV     #6,R1                   ;SET UP MAX # OF DIGITS
005752' 160001                  SUB     R0,R1                   ;GET THE DIFFERENCE
005754' 012703 005776'          MOV     #EBCDWA,R3              ;GET ASCII DATA ADR
005760' 060103                  ADD     R1,R3                   ;INDEX UP TO DESIRED DIGITS
005762' 112302          10$:    MOVB    (R3)+,R2                ;GET DIGIT TO TYPE
005764' 004767 177522           JSR     PC,CHROUT               ;TYPE IT
005770' 005300                  DEC     R0                      ;DONE ALL DIGITS?
005772' 001373                  BNE     10$                     ;Y,N-10$
005774' 000752                  BR      EMESEX                  ;GO TO EXIT

005776' 000003          EBCDWA: .BLKW   3


                                ;****************
                                ;
                                ;  ITOA:  CONVERT BINARY WORD IN REG R3 TO SIX ASCII DIGITS
                                ;         AND TYPE THEM.
                                ;
                                ;       LINK:   JSR     PC,ITOA         S/R CALL
                                ;
                                ;****************

006004' 004467 176706   ITOA:   JSR     R4,SAV04                ;SAVE WORK REG'S
006010' 010300                  MOV     R3,R0                   ;SET WORD UP FOR S/R
006012' 004577 000006'          JSR     R5,@BINASC              ;CONVERT IT TO ASCII
006016' 177760                  .WORD   EBCDWA-.
006020' 004567 001214           JSR     R5,CLIST                ;TYPE THE 6 DIGITS
006024' 177752                  .WORD   EBCDWA-.
006026' 000006                  .WORD   6
006030' 004767 176674   ETYEX: JSR     PC,RST04                ;RESTORE WORK REG'S
006034' 000207                  RTS     PC                      ;EXIT IN-LINE
```

```
1669                                    ;***************
1670                                    ;
1671                                    ;  DATUPK:
1672                                    ;
1673                                    ;       DATE UNPACK AND TYPE S/R
1674                                    ;
1675                                    ;       LINK:   JSR     PC,DATUPK       S/R CALL
1676                                    ;
1677                                    ;       REG R3 CONTAINS PACKED DATE
1678                                    ;
1679                                    ;***************
1680
1681  006036' 004467  176654    DATUPK: JSR     R4,SAV04                ;SAVE WORK REGISTERS
1682  006042' 012704  000105            MOV     #69.,R4                 ;INITIALIZE YEAR TO BASE -1
1683  006046' 042703  100000            BIC     #100000,R3              ;RESET CONTIG BIT
1684  006052' 005204            10$:    INC     R4                      ;ADD 1 TO YEAR VALUE
1685  006054' 162703  001750            SUB     #1000.,R3               ;DECR DATE VALUE BY YEAR INCR
1686  006060' 003374                    BGT     10$                     ;YEAR FOUND? (Y,N-10$)
1687  006062' 062703  001750            ADD     #1000.,R3               ;RESTORE DAY & MONTH
1688  006066' 005046                    CLR     -(SP)                   ;CLEAR LEAP YEAR FLAG
1689  006070' 032704  000003            BIT     #3,R4                   ;THIS A LEAP YEAR?
1690  006074' 001005                    BNE     20$                     ;Y,N-30$
1691  006076' 020327  000073            CMP     R3,#59.                 ;IS DAY LATER THAN FEB 28?
1692  006102' 101402                    BLOS    20$                     ;Y,N-20$
1693  006104' 005216                    INC     (SP)                    ;SET LEAP YEAR FLAG
1694  006106' 005303                    DEC     R3                      ;ADJUST FOR FEB 29
1695  006110' 012700  006217'   20$:    MOV     #EMONTB+3,R0            ;SET UP MONTH TABLE ADR
1696  006114' 111001            30$:    MOVB    (R0),R1                 ;GET # OF DAYS IN CURR MONTH
1697  006116' 020301                    CMP     R3,R1                   ;DAY VALUE =/< CURR MONTH?
1698  006120' 101404                    BLOS    40$                     ;N,Y-40$
1699  006122' 160103                    SUB     R1,R3                   ;DECR DAY VALUE BY CURR MONTH
1700  006124' 062700  000004            ADD     #4,R0                   ;POINT TO NEXT MONTH
1701  006130' 000771                    BR      30$                     ;GO PROCESS NXT MONTH
1702  006132' 005726            40$:    TST     (SP)+                   ;LEAP YEAR FLAG SET?
1703  006134' 001404                    BEQ     50$                     ;Y,N-50$
1704  006136' 020027  006223'           CMP     R0,#EMONTB+7            ;IS MONTH = FEB?
1705  006142' 001001                    BNE     50$                     ;Y,N-50$
1706  006144' 005203                    INC     R3                      ;INCR DAY FOR FEB 29
1707  006146' 004567  177556    50$:    JSR     R5,BCDCV                ;CONVERT & PRINT 2 DAY DIGITS
1708  006152' 000002                    .WORD   2
1709  006154' 012703  006212'           MOV     #EMONMG+4,R3            ;SET UP MSG ADR
1710  006160' 114043                    MOVB    -(R0),-(R3)             ;MOV ASCII MONTH NAME TO MSG
1711  006162' 114043                    MOVB    -(R0),-(R3)
1712  006164' 114043                    MOVB    -(R0),-(R3)
1713  006166' 004567  177462            JSR     R5,MES                  ;PRINT MONTH IN -XXX- FORMAT
1714  006172' 006206'                   .WORD   EMONMG
1715  006174' 010403                    MOV     R4,R3                   ;SET YEAR VALUE IN REG FOR S/R
1716  006176' 004567  177526            JSR     R5,BCDCV                ;CONVERT & PRINT YEAR VALUE
1717  006202' 000002                    .WORD   2
1718  006204' 000711                    BR      ETYEX                   ;GO TO EXIT
1719
1720  006206' 054055  054130  000055 EMONMG: .ASCIZ  /-XXX-/
1721
1722  006214' 040512  017516    EMONTB: .ASCII  /JAN/<31.>
1723  006220' 042506  016102            .ASCII  /FEB/<28.>
1724  006224' 040515  017522            .ASCII  /MAR/<31.>
```

```
1725  006230' 050101  017122                .ASCII  /APR/<30.>
1726  006234' 040515  017531                .ASCII  /MAY/<31.>
1727  006240' 052512  017116                .ASCII  /JUN/<30.>
1729  006244' 052512  017514                .ASCII  /JUL/<31.>
1729  006250' 052501  017507                .ASCII  /AUG/<31.>
1730  006254' 042523  017120                .ASCII  /SEP/<30.>
1731  006260' 041517  017524                .ASCII  /OCT/<31.>
1732  006264' 047516  017126                .ASCII  /NOV/<30.>
1733  006270' 042504  017503                .ASCII  /DEC/<31.>
1734
```

```
                                        .SBTTL  ERROR RETURN POINTS REQUIRED FOR LOAD MEDIA HANDLERS


                                ;***************
                                ;
                                ;       THE FOLLOWING ARE RETURN POINTS FOR SYSTEM DEVICES
                                ;       WHEN THEY ENCOUNTER ERRORS.  THE CURRENT DEVICE
                                ;       FUNCTION WILL BE ABORTED SINCE A RETURN ADR IS
                                ;       NOT SUPPLIED.
                                ;
                                ;       LINK:   JMP     LABEL           WHERE LABEL IS ONE OF
                                ;                                       THE FOLLOWING:
                                ;
                                ;***************

006274' 004567  000162          DELERR: JSR     R5,ERRCOM               ;FILE DELETE ERROR
006300' 042504  020114  051105          .ASCIZ  /DEL ERR/
006306' 000122
                                        .EVEN
006310' 004567  000146          DELFIL: JSR     R5,ERRCOM               ;FILE NEEDS TO BE DELETED ERROR
006314' 042504  020114  046117          .ASCIZ  /DEL OLD/
006322' 000104
                                        .EVEN
006324' 004567  000132          DEVERR: JSR     R5,ERRCOM               ;DEVICE ERROR
006330' 042504  020126  051105          .ASCIZ  /DEV ERR/
006336' 000122
                                        .EVEN
006340' 004567  000116          DEVFUL: JSR     R5,ERRCOM               ;DEVICE IS FULL ERROR
006344' 042504  020126  052506          .ASCIZ  /DEV FULL/
006352' 046114     000
                006356'
                                        .EVEN
006356' 004567  176362          FLNOTF: JSR     R5,BMOVE                ;MOVE FILENAME.EXT TO S/R AREA
006362' 000524'                         .WORD   TXNAM
006364' 002256'                 NFNAME: .WORD   IFNAM
006366' 000011                          .WORD   9.
006370' 004767  177200                  JSR     PC,$TYPNM               ;TYPE NAME OF FILE NOT FOUND
006374' 004567  000062                  JSR     R5,ERRCOM               ;FILE NOT FOUND ERROR
006400' 044506  042514  047040          .ASCIZ  /FILE NOT FND/
006406' 052117  043040  042116
006414'    000
                006416'
                                        .EVEN
006416' 004567  000040          INVCMD: JSR     R5,ERRCOM               ;INVALID CMND FOR DEV
006422' 047111  020126  046503          .ASCIZ  /INV CMND FOR DEV/
006430' 042116  043040  051117
006436' 042040  053105     000
                006444'
                                        .EVEN
006444' 004567  000012          CKSMER: JSR     R5,ERRCOM               ;CHECK SUM ERROR
006450' 045503  052523  020115          .ASCIZ  /CKSUM ERR/
006456' 051105  000122
                                        .EVEN
```

```
                                        ;ERROR COMMON PROCESSING
                                        ;
006462' 004767  176436          ERRCOM: JSR     PC,DELAY                ;DELAY A LITTLE BIT
006466' 004767  000040                  JSR     PC,EXERPT               ;GO REPORT THE ERROR
006472' 005727                          TST     (PC)+                   ;THIS AN ERROR ON A FORCED CLOSE?
006474' 000000                  ECFORC: .WORD   0
006476' 001011                          BNE     10$                     ;N,Y-10$
006500' 005767  173566                  TST     OWCTR                   ;HAS A FILE BEEN LEFT OPEN?
006504' 001410                          BEQ     20$                     ;Y,N-20$
006506' 005267  177762                  INC     ECFORC                  ;SET "FORCING A CLOSE" FLAG
006512' 012705  002354'                 MOV     #OUTDEV,R5              ;SET UP OUTPUT DDB ADR
006516' 004767  176162                  JSR     PC,FCLOSE               ;GO CLOSE THE FILE
006522' 005067  177746          10$:    CLR     ECFORC                  ;RESET FORCED CLOSE FLAG
006526' 000167  174340          20$:    JMP     EXERET                  ;GO TO ERROR EXIT
                                        ;
                                        ;ISSUE STANDARD ERROR MSG S/R
                                        ;
                                        ;JSR    PC,EXERPT       S/R CALL
                                        ;R5 MUST CONTAIN ABS ADR OF ERROR MSG
                                        ;
006532' 010567  000036          EXERPT: MOV     R5,10$                  ;STORE MSG ADR
006536' 105067  171754                  CLRB    PRTFLG                  ;RESET PRINTER/TTY FLAG
006542' 042777  010000  000012'         BIC     #CNTRLO,@CSYSFW         ;RESET TYPING SUPPRESS FLAG
006550' 004567  176170                  JSR     R5,BMOVE                ;SET FUNCTIONAL DEV NAME IN MSG
006554' 006613'                         .WORD   ERMFDN
006556' 006642'                         .WORD   EFDEVN
006560' 000004                          .WORD   4
006562' 004567  177066                  JSR     R5,MES                  ;ISSUE STANDARD PORTION OF
006566' 006604'                         .WORD   EXERM                   ;ERR MSG
006570' 004567  177060                  JSR     R5,MES                  ;ISSUE ERROR I.D.
006574' 000000                  10$:    .WORD   XXXX
006576' 004767  176744                  JSR     PC,CRLF                 ;ISSUE CR/LF
006602' 000207                          RTS     PC                      ;EXIT IN-LINE
                                        ;
006604' 005015  042452  025122  EXERM:  .ASCII  <015><012>/*ER* /
006612' 000040
006613' 000130  054130  020130  ERMFDN: .ASCIZ  /XXXX DEV: /
006620' 042504  035126  000040
006626' 047105  020104  043117  EOFMSG: .ASCIZ  /END OF FILE/
006634' 043040  046111  000105
                                        .EVEN
                                        ;
006642' 047514  042101          EFDEVN: .ASCII  /LOAD/                  ;CURRENT DEVICE'S FUNCTIONAL NAME
```

```
                        .SBTTL  CONSOLE TERMINAL  CONTROL ROUTINES

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

; THESE ROUTINES CONTROL THE ACCESS TO THE CONSOLE TERMINAL
; (READ AND WRITE) FOR BOTH MPG AND USER PROGRAM I/O
; FUNCTIONS. THESE ROUTINES SET UP THE NECESSARY INFORMATION
; REQUIRED BY THE INTERRUPT ROUTINES AND IN THE CASE OF THE
; WRITE, THE I/O TRANSFER IS ALSO INITIATED.
;
; IF THE CONSOLE TERMINAL IS ALREADY BUSY WHEN ENTERED FOR
; AN MPG FUNCTION, THESE ROUTINES WILL AUTOMATICALLY WAIT
; UNTIL THE TERMINAL BECOMES FREE. THE I/O WILL THEN BE
; PERFORMED WITH RETURN BEING MADE IN-LINE ONLY AFTER THE ENTIRE
; I/O FUNCTION HAS BEEN COMPLETED.
;
; FOR USER PROGRAM WRITE FUNCTIONS, RETURN WILL BE MADE TO THE
; SUPPLIED BUSY ADDRESS IF THE TERMINAL IS BUSY UPON ENTRY.
; NO OTHER ACTIONS WILL TAKE PLACE. IF THE TERMINAL IS FREE,
; THE USER I/O WILL BE INITIATED AND THEN THE 'WAIT FOR
; I/O TERM' FLAG WILL BE SET IN THE USER PROGRAM'S FLAGWORD
; AT THE ADDRESS SUPPLIED. RETURN WILL THEN BE MADE IN-LINE.
;
;       MPG LINKAGE:    JSR     R5,CTWR         MPG WRITE
;                         ,CTRD                 MPG READ WITH ECHO
;                         ,CTRDNE               MPG READ WITHOUT ECHO
;                       .WORD   TAG-.           RELATIVE ADR OF I/O AREA
;                       .WORD   COUNT           # OF CHARACTERS TO TRANSFER
;
;
;       USER LINKAGE:   JSR     R5,CTUSWR       USER WRITE
;                       .WORD   TAG-.           RELATIVE ADR OF I/O AREA
;                       .WORD   COUNT           # OF CHARS TO XFER
;                       .WORD   BADR-.          RELATIVE RETURN ADR IF BUSY
;                       .WORD   FWADR           ABSOLUTE ADR OF PROG'S FLGWD
;
;       DESTROYS REG'S: R0,R1,R2
;
; RETURN WILL BE IN-LINE FOLLOWING THE ARGUMENTS UNLESS THE CONSOLE
; TERMINAL IS BUSY AND A USER I/O IS BEING ATTEMPTED.
;
;*****************************************************************************

                        ;USER WRITE ENTRY POINT

006646' 032777 010000 000012' CTUSWR: BIT     #CNTRLO,@CSYSFW         ;IS TYPING SUPPRESSED?
006654' 001421                        BEQ     30$                     ;Y,N-30$
006656' 032777 040000 000012'         BIT     #CRUDED,@CSYSFW         ;SUPPOSED TO RESET USER DEDICATED?
006664' 001412                        BEQ     20$                     ;Y,N-20$
006666' 042777 060000 000012'         BIC     #CRUDED+CTUDED,@CSYSFW  ;RESET SYSTEM DEDICATED FLGS
006674' 016567 000006 002554          MOV     6(R5),UFWHLD            ;GET FLAGWORD ADR
006702' 004577 000002'                JSR     R5,@UFWCLR              ;RESET USER'S DEDICATED FLAG
006706' 000040                        .WORD   SETDED
006710' 002546                        .WORD   UFWHLD-.
006712' 062705 000010         20$:    ADD     #8.,R5                  ;SET UP RETURN ADR
```

```
1876  006716' 000205                                  RTS     R5                              ;EXIT IN-LINE
1877  006720' 032767  002042  002130  30$:    BIT     #CTWBSY+CTEBSY+CTWAIT,CTFLGW    ;WR BUSY OR ECHO BUSY SET?
1878  006726' 001404                          BEQ     CTDUWR                          ;(Y,N-CTDUWR)
1879  006730' 062705  000004          CTBRET: ADD     #4,R5                           ;INCR RET ADR PAST ARGUMENTS
1880  006734' 061505                          ADD     (R5),R5                         ;SET UP USER BUSY RET ADR
1881  006736' 000205                          RTS     R5                              ;EXIT TO USER'S BUSY ADR
1882  006740' 052767  100002  002110  CTDUWR: BIS     #CTWUSR+CTWBSY,CTFLGW           ;SET USER WR & WR BUSY FLAGS
1883  006746' 000421                          BR      CTWCOM                          ;GO TO COMMON PROCESSING
1884
1885                                          ;EXEC WRITE ENTRY POINT
1886
1887  006750' 032777  010000  000012' CTWR:   BIT     #CNTRLO,@CSYSFW                 ;IS TYPING SUPPRESSED?
1888  006756' 001403                          BEQ     40$                             ;Y,N-40$
1889  006760' 062705  000004                  ADD     #4,R5                           ;SET UP RETURN ADR
1890  006764' 000205                          RTS     R5                              ;EXIT IN-LINE
1891  006766' 032767  002042  002062  40$:    BIT     #CTWBSY+CTEBSY+CTWAIT,CTFLGW    ;WR BUSY OR ECHO BUSY FLAGS SET?
1892  006774' 001365                          BNE     CTWR                            ;(N,Y-CTWR)
1893  006776' 042767  100000  002052          BIC     #CTWUSR,CTFLGW                  ;RESET USER WRITE FLAG
1894  007004' 052767  000002  002044          BIS     #CTWBSY,CTFLGW                  ;SET THE WRITE BUSY FLAG
1895  007012' 010500                  CTWCOM: MOV     R5,R0                           ;GET DATA ADR INDEX VALUE AND
1896  007014' 062500                          ADD     (R5)+,R0                        ;ADJUST TO ABSOLUTE ADR
1897  007016' 010067  002040                  MOV     R0,CTWADR                       ;STORE DATA ADR FOR INT ROUTINE
1898  007022' 012567  002032                  MOV     (R5)+,CTWCNT                    ;GET & STORE CHAR CNT
1899  007026' 005767  002024                  TST     CTFLGW                          ;DOING A USER WRITE?
1900  007032' 100007                          BPL     2$                              ;(Y,N-2$)
1901  007034' 005725                          TST     (R5)+                           ;INCR PAST BUSY ADR
1902  007036' 012567  002022                  MOV     (R5)+,CTWUFW                    ;MOVE USER FLGWD ADR TO INT ROUT
1903  007042' 004577  000000'                 JSR     R5,@UFWSET                      ;SET USER'S WAIT FOR
1904  007046' 000020                          .WORD   CTPRIO                          ;CONSOLE/PRINTER I/O FLAG
1905  007050' 002014                          .WORD   CTWUFW-.
1906  007052' 052777  000100  002066  2$:     BIS     #100,@CTWDRA                    ;SET WRITE INT ENABLE BIT
1907  007060' 005767  001772                  TST     CTFLGW                          ;IS THIS A USER WRITE?
1908  007064' 100406                          BMI     4$                              ;(N,Y-4$)
1909  007066' 004767  171306          3$:     JSR     PC,LIGHTS                       ;DISPLAY IDLE BIT PATTERN
1910  007072' 032767  000002  001756          BIT     #CTWBSY,CTFLGW                  ;IS 'WR BUSY' FLAG STILL SET?
1911  007100' 001372                          BNE     3$                              ;(N,Y-3$)
1912  007102' 000205                  4$:     RTS     R5                              ;EXIT IN-LINE
1913
1914
1915
1916                                          ;EXEC READ WITHOUT ECHO ENTRY POINT
1917
1918  007104' 032767  000003  001744  CTRDNE: BIT     #CTRBSY+CTWBSY,CTFLGW           ;READ BUSY OR WRITE BUSY?
1919  007112' 001374                          BNE     CTRDNE                          ;(N,Y-CTRDNE)
1920  007114' 042767  000200  001734          BIC     #CTRDEC,CTFLGW                  ;RESET THE ECHO FLAG
1921  007122' 000407                          BR      CTRCOM                          ;GO TO COMMON PROCESSING
1922
1923                                          ;EXEC READ WITH ECHO ENTRY POINT
1924
1925  007124' 032767  000003  001724  CTRD:   BIT     #CTRBSY+CTWBSY,CTFLGW           ;READ BUSY OR WRITE BUSY?
1926  007132' 001374                          BNE     CTRD                            ;(N,Y-CTRD)
1927              001                         .IF DF MIMIC
1928                                          BR      CTRDNE                          ;DO NOT ECHO IF MIMIC
1929              000                         .ENDC
1930  007134' 052767  000200  001714          BIS     #CTRDEC,CTFLGW                  ;SET THE ECHO FLAG
1931  007142' 042767  000040  001706  CTRCOM: BIC     #CTEBSY,CTFLGW                  ;RESET THE ECHO BUSY FLAG
```

```
1932  007150' 010500                          MOV     R5,R0                   ;GET DATA ADR INDEX VALUE AND
1933  007152' 062500                          ADD     (R5)+,R0                ;ADJUST TO AN ABSOLUTE ADR
1934  007154' 012501                          MOV     (R5)+,R1                ;GET CHARACTER COUNT
1935  007156' 012702  011066'                 MOV     #CTRCNT,R2              ;SET UP ADR OF RD STORAGE AREA
1936  007162' 010122                          MOV     R1,(R2)+                ;STORE COUNT AND DATA ADR
1937  007164' 010022                          MOV     R0,(R2)+                ;IN THE INT ROUT
1938  007166' 010122                          MOV     R1,(R2)+
1939  007170' 010022                          MOV     R0,(R2)+
1940  007172' 105767  001660                  TSTB    CTFLGW                  ;ECHO FLAG SET?
1941  007176' 100003                          BPL     12$                     ;(Y,N-12$)
1942  007200' 052767  000040  001650          BIS     #CTEBSY,CTFLGW          ;SET THE ECHO BUSY FLAG
1943  007206' 052767  000001  001642  12$:    BIS     #CTRBSY,CTFLGW          ;SET THE READ BUSY FLAG
1944  007214' 012777  000100  001716          MOV     #100,@CTRDRA            ;SET TTY RD INT ENABLE
1945              001                         .IF DF MIMIC
1946                                          .IFT
1947                                  CTRMLP: MOV     #12,R0          ;*** MIMIC ***
1948                                          .IFTF
1949  007222' 004767  171152          CTRTRB: JSR     PC,LIGHTS               ;DISPLAY IDLE BIT PATTERN
1950  007226' 032767  000001  001622          BIT     #CTRBSY,CTFLGW          ;IS READ BUSY STILL SET?
1951                                          .IFF
1952  007234' 001372                          BNE     CTRTRB                  ;(N,Y-CTRTRB)
1953                                          .IFT
1954                                          BNE     CTRMCD          ;*** MIMIC ***
1955                                          .IFTF
1956  007236' 000205                          RTS     R5                      ;EXIT IN-LINE
1957                                          .IFT
1958
1959                                  CTRMCD: DEC     R0              ;*** MIMIC ***
1960                                          BNE     CTRTRB          ;*** MIMIC ***
1961                                  SLPBKP: NOP                     ;*** MIMIC ***
1962                                          BR      CTRMLP          ;*** MIMIC ***
1963              000                         .ENDC
```

```
                                .SBTTL  LIST DEVICE CONTROL ROUTINE

                        ;*****************************************************************
                        ;
                        ;  THIS ROUTINE DETERMINES IF A LIST DEVICE HAS BEEN ASSIGNED
                        ;  AND THEN BRANCHES TO EITHER THE CONSOLE TERMINAL ROUTINE
                        ;  OR TO THE PRINTER ROUTINE.  IF THE PRINTER/TTY FLAG IS RESET
                        ;  IT WILL ALWAYS GO TO THE CONSOLE TERMINAL.
                        ;
                        ;  LINKAGE:     JSR     R5,CLIST        ROUTINE CALL
                        ;               .WORD   MSGADR-.        REL ADR OF MSG DATA
                        ;               .WORD   COUNT           MSG BYTE COUNT
                        ;
                        ;  EXITS TO EITHER CTWR OR CPRT
                        ;
                        ;*****************************************************************

007240' 105767  171252  CLIST:  TSTB    PRTFLG                  ;PRINTER/TTY FLAG SET?
007244' 001641                  BEQ     CTWR                    ;Y,N-CTWR
007246' 005767  002170          TST     CLISTI+4                ;A PRINTER BEEN ASSIGNED?
007252' 001636                  BEQ     CTWR                    ;Y,N-CTWR
007254' 000400                  BR      CPRT                    ;GO TO PRINTER ROUTINE
```

```
                                        .SBTTL  PRINTER CONTROL ROUTINE

                                ;*****************************************************************
                                ;
                                ;  THIS ROUTINE CONTROLS WRITE OPERATIONS TO THE MPG LIST
                                ;  DEVICE FOR BOTH MPG AND USER FUNCTIONS.  IF AN MPG
                                ;  FUNCTION, THIS ROUTINE WILL WAIT FOR THE PRINTER IF IT IS
                                ;  ALREADY BUSY UPON ENTRY AND WILL WAIT UNTIL THE PRINTER
                                ;  COMPLETES THE SPECIFIED OPERATION BEFORE RETURNING.
                                ;
                                ;  FOR USER FUNCTIONS, RETURN WILL BE MADE TO THE SUPPLIED BUSY
                                ;  ADDRESS IF THE PRINTER IS BUSY UPON ENTRY.  NO OTHER ACTIONS WILL
                                ;  TAKE PLACE.  IF THE PRINTER IS FREE, THE USER'S WRITE WILL BE
                                ;  INITIATED AND THEN THE 'WAIT FOR I/O TERM' FLAG WILL BE SET
                                ;  IN THE USER PROGRAM'S FLAGWORD AT THE ADDRESS SUPPLIED.  RETURN
                                ;  WILL THEN BE IMMEDIATELY MADE IN-LINE.
                                ;
                                ;  MPG LINKAGE: JSR     R5,CPRT         MPG PRINT
                                ;               .WORD   TAG-.           REL ADR OF I/O AREA
                                ;               .WORD   COUNT           # OF CHARS TO TRANSFER
                                ;
                                ;  USER LINKAGE:JSR     R5,CUSPRT       USER PRINT
                                ;               .WORD   TAG-.           REL ADR OF I/O AREA
                                ;               .WORD   COUNT           # OF CHARS TO XFER
                                ;               .WORD   BADR-.          REL RETURN ADR IF BUSY
                                ;               .WORD   FWADR           ABSOLUTE ADR OF PROG'S FLGWD
                                ;
                                ;  DESTROYS REGISTERS: R0,R1,R2
                                ;
                                ;*****************************************************************


                                        ;MPG ENTRY POINT

007256' 032777  000100  002162  CPRT:   BIT     #100,@PRTDRA            ;IS THE PRINTER BUSY?
007264' 001404                          BEQ     1$                      ;Y,N-1$
007266' 032777  004000  000012'         BIT     #CHPINT,@CSYSFW         ;INT ENABLE BEING HELD SET?
007274' 001770                          BEQ     CPRT                    ;Y,N-CPRT
007276' 042767  100000  002122  1$:     BIC     #CPRUSR,CPRFWD          ;RESET USER PRINT FLAG
007304' 000432                          BR      CPRCOM                  ;GO TO COMMON PROCESSING

                                        ;USER ENTRY POINT

007306' 032777  000100  002132  CUSPRT: BIT     #100,@PRTDRA            ;PRINTER BUSY?
007314' 001423                          BEQ     5$                      ;Y,N-5$
007316' 032777  004000  000012'         BIT     #CHPINT,@CSYSFW         ;INT ENABLE BEING HELD SET?
007324' 001415                          BEQ     3$                      ;Y,N-3$
007326' 016567  000006  002122          MOV     6(R5),UFWHLD            ;GET FLAGWORD ADR
007334' 004577  000004'                 JSR     R5,@UFWTST              ;BEING HELD SET FOR THIS USER?
007340' 000040                          .WORD   SETDED
007342' 002114                          .WORD   UFWHLD-.
007344' 000405                          BR      3$                      ;Y,N-3$
007346' 004577  000004'                 JSR     R5,@UFWTST              ;USER'S PREV PRINT COMPLETED?
007352' 000020                          .WORD   CTPRIO
007354' 002102                          .WORD   UFWHLD-.
007356' 000402                          BR      5$                      ;N,Y-5$
```

```
2044  007360' 000167 177344          3$:     JMP     CTBRET          ;GO TO BUSY RETURN
2045  007364' 052767 100000 002034   5$:     BIS     #CPRUSR,CPRFWD  ;SET USER PRINT FLAG
2046  007372' 012702 011426'         CPRCOM: MOV     #CPRFWD,R2      ;SET UP FLAGWORD ADR
2047  007376' 105777 002044          10$:    TSTB    @PRTDRA         ;PRINTER READY SET?
2048  007402' 100003                         BPL     15$             ;Y,N-15$
2049  007404' 005777 002036                  TST     @PRTDRA         ;PRINTER ERROR?
2050  007410' 100021                         BPL     30$             ;Y,N-30$
2051  007412' 105712                 15$:    TSTB    (R2)            ;PRINTER ERROR FLAG SET?
2052  007414' 100413                         BMI     20$             ;N,Y-20$
2053  007416' 052712 000200                  BIS     #CPRERR,(R2)    ;SET ERROR FLAG
2054  007422' 010246                         MOV     R2,-(SP)        ;SAVE FLGWD ADR
2055  007424' 042777 010000 000012'         BIC     #CNTRLO,@CSYSFW ;RESET SUPPRESS TYPING FLAG
2056  007432' 004567 177312                  JSR     R5,CTWR         ;ISSUE PRINTER ERROR MSG
2057  007436' 000142                         .WORD   CPRTER-.
2058  007440' 000026                         .WORD   22.
2059  007442' 012602                         MOV     (SP)+,R2        ;RESTORE FLGWD ADR
2060  007444' 005712                 20$:    TST     (R2)            ;THIS A USER PRINT?
2061  007446' 100353                         BPL     10$             ;Y,N-10$
2062  007450' 000167 177254                  JMP     CTBRET          ;GO TAKE BUSY RETURN
2063  007454' 112712 000001          30$:    MOVB    #1,(R2)         ;RESET PRT ERR & SET PRT BSY FLGS
2064  007460' 012700 011430'                 MOV     #CPRADR,R0      ;SET UP ADR OF STORAGE AREA
2065  007464' 010510                         MOV     R5,(R0)         ;GET DATA ADR & MAKE IT
2066  007466' 062520                         ADD     (R5)+,(R0)+     ;ABSOLUTE
2067  007470' 012520                         MOV     (R5)+,(R0)+     ;STORE CHAR CNT
2068  007472' 005712                         TST     (R2)            ;USER PRINT?
2069  007474' 100015                         BPL     40$             ;Y,N-40$
2070  007476' 005725                         TST     (R5)+           ;INCR PAST BUSY RET ADR
2071  007500' 012510                         MOV     (R5)+,(R0)      ;STORE USER'S FLGWD ADR
2072  007502' 004577 000000'                 JSR     R5,@UFWSET      ;SET USER'S WAIT FOR
2073  007506' 000020                         .WORD   CTPRIO          ;CONSOLE/PRINTER I/O FLAG
2074  007510' 001724                         .WORD   CPRUFW-.
2075  007512' 032777 020000 000012'         BIT     #CTUDED,@CSYSFW ;PRINTER DEDICATED TO USER?
2076  007520' 001403                         BEQ     40$             ;Y,N-40$
2077  007522' 052777 004000 000012'         BIS     #CHPINT,@CSYSFW ;SET 'HOLD INT ENB SET' FLAG
2078  007530' 016701 001714          40$:    MOV     PRTIVA,R1       ;GET PRINTER'S INT VEC ADR
2079  007534' 012721 011162'                 MOV     #CPRINT,(R1)+   ;STORE INT ROUT ADR AT VECTOR
2080  007540' 016711 001706                  MOV     PRTPSW,(R1)     ;STORE PS WORD AT VECTOR
2081  007544' 005077 001676                  CLR     @PRTDRA         ;MAY HAVE TO TOGGLE INT ENABLE
2082  007550' 012777 000100 001670           MOV     #100,@PRTDRA    ;SET PRINTER'S INT ENABLE
2083  007556' 005712                         TST     (R2)            ;THIS A USER PRINT?
2084  007560' 100406                         BMI     60$             ;N,Y-60$
2085  007562' 004767 170612          50$:    JSR     PC,LIGHTS       ;DISPLAY IDLE BIT PATTERN
2086  007566' 032767 000001 001632           BIT     #CPRBSY,CPRFWD  ;PRINTER STILL BUSY?
2087  007574' 001372                         BNE     50$             ;N,Y-50$
2088  007576' 000205                 60$:    RTS     R5              ;EXIT TO IN-LINE ADR
2089
2090  007600' 042452 025122 046040   CPRTER: .ASCII  /*ER* LST DEV: HW ERR/<015><012>
      007606' 052123 042040 053105
      007614' 020072 053510 042440
      007622' 051122 005015
2091                                   .EVEN
```

```
                                        .SBTTL  CONSOLE TERMINAL INTERRUPT ROUTINES


                                        ;CONSOLE TERMINAL WRITE INTERRUPT

007626' 004567 001172           CTWINT: JSR     R5,CSVREG               ;GO SAVE ALL REGISTERS
007632' 016702 001310                   MOV     CTWDRA,R2               ;SET UP DEV REG'S ADR
007636' 005722                          TST     (R2)+                   ;ADJUST IT TO DEV BUF ADR
007640' 012705 011076'                  MOV     #CTFTCT,R5              ;SET UP TAB/FILL COUNT ADR
007644' 005715                          TST     (R5)                    ;TAB/FILL COUNT ZERO?
007646' 001403                          BEQ     CTWRCE                  ;(N,Y-CTWRCE)
007650' 005325                          DEC     (R5)+                   ;DECR TAB/FILL CNT BY 1
007652' 011512                  CTWFOT: MOV     (R5),(R2)               ;MOVE FILL/SPACE CHAR TO DEV BUF
007654' 000454                          BR      CTWREX                  ;GO TO INT ROUT EXIT
007656' 012704 011106'          CTWRCE: MOV     #CTECNT,R4              ;SET UP ADR OF ECHO BUF CNTRL WDS
007662' 005724                          TST     (R4)+                   ;ECHO BUF CNT ZERO?
007664' 001411                          BEQ     CTWRCD                  ;(N,Y-CTWRCD)
007666' 021464 000002                   CMP     (R4),2(R4)              ;AT THE END OF THE BUF?
007672' 001002                          BNE     1$                      ;(Y,N-1$)
007674' 162714 000024                   SUB     #24,(R4)                ;SET UP BUF START ADR
007700' 113400                  1$:     MOVB    @(R4)+,R0               ;GET CHAR TO ECHO
007702' 005244                          INC     -(R4)                   ;ADD 1 TO CURR BUF ADR
007704' 005344                          DEC     -(R4)                   ;SUB 1 FROM BUF COUNT
007706' 000456                          BR      CTCKCH                  ;GO CHECK THIS CHARACTER
007710' 012703 011060'          CTWRCD: MOV     #CTWCNT,R3              ;SET UP ADR OF DATA COUNT
007714' 005713                          TST     (R3)                    ;DATA CHAR COUNT ZERO?
007716' 001036                          BNE     CTDODT                  ;(Y,N-CTDODT)
007720' 042742 000100           CTWABT: BIC     #100,-(R2)              ;RESET WRITE'S INT ENABLE
007724' 032743 000040                   BIT     #CTEBSY,-(R3)           ;WERE WE DOING ECHO'S?
007730' 001403                          BEQ     3$                      ;(Y,N-3$)
007732' 004567 000770                   JSR     R5,CTRTRM               ;GO TERMINATE THE RD FUNCTION
007736' 000423                          BR      CTWREX                  ;GO TO INT ROUT EXIT
007740' 042713 000002           3$:     BIC     #CTWBSY,(R3)            ;RESET WR BSY FLAG
007744' 005713                          TST     (R3)                    ;THIS A USER WRITE?
007746' 100017                          BPL     CTWREX                  ;(Y,N-CTWREX)
007750' 004577 000002'                  JSR     R5,@UFWCLR              ;RESET USER'S WAIT FOR
007754' 000020                          .WORD   CTPRIO                  ;CONSOLE/PRINTER I/O FLAG
007756' 001106                          .WORD   CTWUFW-.
007760' 032777 040000 000012'           BIT     #CRUDED,@CSYSFW         ;SUPPOSED TO RESET USER DEDICATED?
007766' 001407                          BEQ     CTWREX                  ;Y,N-CTWREX
007770' 042777 060000 000012'           BIC     #CTUDED+CRUDED,@CSYSFW  ;RESET SYSTEM DEDICATED FLAGS
007776' 004577 000002'                  JSR     R5,@UFWCLR              ;RESET USER DEDICATED FLAG
010002' 000040                          .WORD   SETDED
010004' 001060                          .WORD   CTWUFW-.
010006' 004567 001026           CTWREX: JSR     R5,CRTREG               ;GO RESTORE REGISTERS
010012' 000002                          RTI                             ;RETURN FROM INTERRUPT
010014' 032743 001000           CTDODT: BIT     #CTCREQ,-(R3)           ;CONSOLE REQUEST PENDING?
010020' 001403                          BEQ     5$                      ;(Y,N-5$)
010022' 005723                          TST     (R3)+                   ;POINT AT WRITE COUNT
010024' 005013                          CLR     (R3)                    ;CLEAR WRITE DATA COUNT
010026' 000734                          BR      CTWABT                  ;GO ABORT THIS WRITE
010030' 117300 000004           5$:     MOVB    @4(R3),R0               ;GET DATA CHARACTER
010034' 005263 000004                   INC     4(R3)                   ;INCREMENT DATA ADR
010040' 005363 000002                   DEC     2(R3)                   ;DECREMENT DATA COUNT
010044' 062705 000004           CTCKCH: ADD     #4,R5                   ;ADJUST FILL/TAB TABLE ADR
010050' 120027 000040                   CMPB    R0,#40                  ;THIS A CONTROL/SPECIAL CHAR?
```

```
2149  010054' 103023                      BHIS    8$                      ;(Y,N-8$)
2150  010056' 122700 000011               CMPB    #011,R0                 ;IS CHAR A TAB?
2151  010062' 001010                      BNE     7$                      ;(Y,N-7$)
2152  010064' 005315                      DEC     (R5)                    ;DECR CURR TAB CNT BY 1
2153  010066' 011565 177774               MOV     (R5),-4(R5)             ;MOVE CURR TAB CNT TO TAB/FILL CNT
2154  010072' 012715 000010               MOV     #10,(R5)                ;RESTORE CURR TAB COUNT
2155  010076' 012745 000040               MOV     #40,-(R5)               ;MOVE A SPACE TO FILL/SPACE CHAR
2156  010102' 000663                      BR      CTWFOT                  ;GO BEGIN ISSUING SPACES
2157  010104' 122700 000015       7$:     CMPB    #015,R0                 ;IS IT A CARRIAGE RETURN?
2158  010110' 001407                      BEQ     9$                      ;(N,Y-9$)
2159  010112' 122700 000033               CMPB    #033,R0                 ;IS IT AN ALTMODE?
2160  010116' 001006                      BNE     11$                     ;(Y,N-11$)
2161  010120' 112700 000044               MOVB    #044,R0                 ;SET UP "$" CHARACTER
2162  010124' 005315               8$:     DEC     (R5)                    ;DECR CURR TAB CNT BY 1
2163  010126' 001002                      BNE     11$                     ;CNT = ZERO?  (Y,N-11$)
2164  010130' 012715 000010       9$:     MOV     #10,(R5)                ;RESTORE CURR TAB CNT TO 8
2165  010134' 120067 001020       11$:    CMPB    R0,CTFID                ;FILLS AFTER THIS CHAR?
2166  010140' 001006                      BNE     13$                     ;(Y,N-13$)
2167  010142' 016767 001006 000726        MOV     CTFCNT,CTFTCT           ;SET UP FILL COUNT
2168  010150' 016767 001002 000722        MOV     CTFCAR,CTFSCH           ;SET UP FILL CHAR
2169  010156' 110012               13$:    MOVB    R0,(R2)                 ;MOVE CHAR TO DEVICE BUF
2170  010160' 000712                      BR      CTWREX                  ;GO TO INT ROUT EXIT
2171
2172
2173
2174                                      ;CONSOLE TERMINAL READ INTERRUPT ROUTINE
2175
2176  010162' 004567 000636       CTRINT: JSR     R5,CSVREG               ;GO SAVE ALL REGISTERS
2177  010166' 012704 011056'              MOV     #CTFLGW,R4              ;SET UP ADR OF CONS TERM FLGWD
2178  010172' 016702 000742               MOV     CTRDRA,R2               ;SET UP DEV REG'S ADR
2179  010176' 016200 000002               MOV     2(R2),R0                ;GET READ DATA CHARACTER
2180  010202' 042700 000200               BIC     #200,R0                 ;RESET HIGH ORDER BIT
2181  010206' 120027 000141       LOWCAS: CMPB    R0,#141                 ;THIS A LOWER CASE ALPHA?
2182  010212' 103405                      BLO     10$                     ;Y,N-10$
2183  010214' 120027 000172               CMPB    R0,#172
2184  010220' 101002                      BHI     10$
2185  010222' 042700 000040               BIC     #40,R0                  ;RESET LOWER CASE BIT
2186  010226' 120027 000023       10$:    CMPB    R0,#023                 ;THIS A CNTRL/S?
2187  010232' 001003                      BNE     20$                     ;Y,N-20$
2188  010234' 052714 002000               BIS     #CTWAIT,(R4)            ;SET SOFTWARE "WAIT" FLAG
2189  010240' 000503                      BR      CTRDEX                  ;GO TO INT EXIT
2190  010242' 032714 002000       20$:    BIT     #CTWAIT,(R4)            ;SOFTWARE "WAIT" FLAG SET?
2191  010246' 001403                      BEQ     30$                     ;Y,N-30$
2192  010250' 042714 002000               BIC     #CTWAIT,(R4)            ;RESET THE "WAIT" FLAG
2193  010254' 000475                      BR      CTRDEX                  ;GO TO INT EXIT
2194  010256' 120027 000017       30$:    CMPB    R0,#017                 ;THIS A CNTRL/O?
2195  010262' 001004                      BNE     40$                     ;Y,N-40$
2196  010264' 052777 010000 000012'       BIS     #CNTRLO,@CSYSFW         ;SET THE SUPPRESS TYPING FLAG
2197  010272' 000466                      BR      CTRDEX                  ;GO TO INT EXIT
2198  010274' 032714 001000       40$:    BIT     #CTCREQ,(R4)            ;IS CONSOLE REQUEST FLAG SET?
2199  010300' 001063                      BNE     CTRDEX                  ;(N,Y-CTRDEX)
2200  010302' 120027 000003               CMPB    R0,#003                 ;THIS CHAR A CNTRL/C?
2201  010306' 001527                      BEQ     CTRLC                   ;(N,Y-CTRLC)
2202  010310' 032714 000001               BIT     #CTRBSY,(R4)            ;IS RD BUSY FLAG SET?
2203  010314' 001455                      BEQ     CTRDEX                  ;(Y,N-CTRDEX)
2204  010316' 012705 011070'              MOV     #CTRADR,R5              ;SET UP ADR OF READ INFO
```

```
2205  010322' 120027 000025          CMPB   R0,#025          ;IS IT A CTRL/U?
2206  010326' 001502                 BEQ    CTRCTU           ;(N,Y-CTRCTU)
2207  010330' 122700 000177          CMPB   #177,R0          ;RD DATA A RUBOUT?
2208  010334' 001450                 BEQ    CTRRKY           ;(N,Y-CTRRKY)
2209  010336' 032714 000100          BIT    #CTRRUB,(R4)     ;RUBOUT FLAG SET?
2210  010342' 001405                 BEQ    50$              ;(Y,N-50$)
2211  010344' 004567 000376          JSR    R5,CTRPEB        ;PUT A BACKSLASH IN ECHO BUF
2212  010350'    134    000          .BYTE  134,0
2213  010352' 042714 000100          BIC    #CTRRUB,(R4)     ;RESET RUBOUT FLAG
2214  010356' 110035          50$:   MOVB   R0,@(R5)+        ;MOVE RD DATA CHAR TO USER ADR
2215  010360' 005245                 INC    -(R5)            ;INCR RD DATA ADR BY 1
2216  010362' 122700 000015          CMPB   #015,R0          ;IS THIS A CARR RET?
2217  010366' 001004                 BNE    52$              ;(Y,N-52$)
2218             001                 .IF DF MIMIC
2219                                 .IFF
2220  010370' 004567 000352  51$:    JSR    R5,CTRPEB        ;PUT CR/LF IN ECHO BUF
2221  010374'    015    012          .BYTE  015,012
2222                                 .IFT
2223                          51$:   MOV    #5015,R3         ;SET UP CR/LF DATA
2224                                 JSR    R5,CTWPEB        ;PUT THEM IN THE ECHO BUF
2225             000                 .ENDC
2226  010376' 000416                 BR     54$              ;GO PROCESS RD END
2227  010400' 122700 000012  52$:    CMPB   #012,R0          ;IS IT A LINE FEED?
2228  010404' 001771                 BEQ    51$              ;N,Y-51$
2229  010406' 110067 000004          MOVB   R0,53$           ;SET UP RD DATA FOR ECHO S/R
2230  010412' 004567 000330          JSR    R5,CTRPEB        ;PUT CHAR IN ECHO BUF
2231  010416' 000000          53$:   .WORD  XXXX
2232  010420' 005345                 DEC    -(R5)            ;DECR DATA COUNT BY ONE
2233  010422' 001012                 BNE    CTRDEX           ;DATA CNT = 0? (Y,N-CTRDEX)
2234  010424' 112775 000015 000002  MOVB   #015,@2(R5)       ;STORE CARR RET IN USER'S AREA
2235             001                 .IF DF MIMIC
2236                                 .IFF
2237  010432' 000756                 BR     51$              ;GO ECHO CR/LF & TERMINATE READ
2238                                 .IFT
2239                                 NOP                     ;REPLACE CODE IS 000755
2240             000                 .ENDC
2241  010434' 052714 000400  54$:    BIS    #CTRDON,(R4)     ;SET READ DONE FLAG
2242  010440' 105714                 TSTB   (R4)             ;IS ECHO FLAG SET?
2243  010442' 100402                 BMI    CTRDEX           ;(N,Y-CTRDEX)
2244  010444' 004567 000256          JSR    R5,CTRTRM        ;GO TERMINATE THIS READ
2245  010450' 004567 000364  CTRDEX: JSR    R5,CRTREG        ;RESTORE ALL REG'S
2246  010454' 000002                 RTI                     ;RETURN AFTER INTERRUPT
2247
2248                                          ;RUBOUT KEY PROCESSING
2249
2250  010456' 032714 000100  CTRRKY: BIT    #CTRRUB,(R4)     ;RUBOUT FLAG ALREADY SET?
2251  010462' 001005                 BNE    56$              ;(N,Y-56$)
2252  010464' 052714 000100          BIS    #CTRRUB,(R4)     ;SET THE RUBOUT FLAG
2253  010470' 004567 000252          JSR    R5,CTRPEB        ;GO PUT BACKSLASH IN ECHO BUF
2254  010474'    134    000          .BYTE  134,0
2255  010476' 021565 000004  56$:    CMP    (R5),4(R5)       ;CURR DATA ADR = ORG DATA ADR?
2256  010502' 001004                 BNE    58$              ;(Y,N-58$)
2257  010504' 004567 000236          JSR    R5,CTRPEB        ;SET UP BACKSLASH FOR ECHO
2258  010510'    134    000          .BYTE  134,0
2259  010512' 000417                 BR     CTRECL           ;GO DO CARR RET & LINE FEED
2260  010514' 005325          58$:   DEC    (R5)+            ;DECR DATA ADR BY 1
```

```
2261  010516' 115567 000006          MOVB   @-(R5),7O$          ;GET CHAR THAT IS BEING DELETED
2262  010522' 005245                 INC    -(R5)               ;INCREMENT DATA COUNT BY 1
2263  010524' 004567 000216          JSR    R5,CTRPEB           ;GO SET UP DELETED CHAR FOR ECHO
2264  010530' 000000          7O$:   .WORD  0
2265  010532' 000746                 BR     CTRDEX              ;GO TO INT ROUT EXIT
2266
2267                                                ;CONTROL 'U' PROCESSING
2268
2269  010534' 016515 000004  CTRCTU: MOV    4(R5),(R5)          ;SET UP ORG DATA ADR
2270  010540' 016545 000002          MOV    2(R5),-(R5)         ;SET UP ORG DATA COUNT
2271  010544' 004567 000176          JSR    R5,CTRPEB           ;PUT "↑" AND "U" IN ECHO BUF
2272  010550'    136    125          .BYTE  136,125
2273  010552' 004567 000170  CTRECL: JSR    R5,CTRPEB           ;PUT "CR/LF" IN ECHO BUF
2274  010556'    015    012          .BYTE  015,012
2275  010560' 042714 000100          BIC    #CTRRUB,(R4)        ;RESET RUBOUT FLAG
2276  010564' 000731                 BR     CTRDEX              ;GO TO INT ROUT EXIT
2277
2278                                                ;CONTROL "C" PROCESSING
2279
2280  010566' 005777 000012' CTRLC:  TST    @CSYSFW             ;CONSOLE REQ BEING PROCESSED?
2281  010572' 100726                 BMI    CTRDEX              ;(N,Y-CTRDEX)
2282  010574' 032777 002000 000012' BIT    #CUSPGR,@CSYSFW     ;USER PROG RUNNING?
2283  010602' 001722                 BEQ    CTRDEX              ;(Y,N-CTRDEX)
2284  010604' 032714 000001          BIT    #CTRBSY,(R4)        ;IS RD BUSY FLG SET?
2285  010610' 001404                 BEQ    CTRCRQ              ;Y,N-CTRCRQ
2286  010612' 052714 000400          BIS    #CTRDON,(R4)        ;SET READ DONE FLAG
2287  010616' 004567 000104          JSR    R5,CTRTRM           ;GO TERMINATE READ FUNCTION
2288
2289                                                ;CONSOLE REQUEST RECEIVED
2290
2291  010622' 012703 041536  CTRCRQ: MOV    #"↑C,R3             ;SET UP "↑" & "C" CHARS
2292  010626' 004567 000122          JSR    R5,CTWPEB           ;GO PUT THEM IN ECHO BUFFER
2293  010632' 052714 001000          BIS    #CTCREQ,(R4)        ;SET THE CONSOLE REQ FLAG
2294  010636' 012746 000022          MOV    #18.,-(SP)          ;SET UP STK PNTR ADR
2295  010642' 060616                 ADD    SP,(SP)             ;AT TIME OF INT
2296  010644' 032777 000001 000012' BIT    #MMVER,@CSYSFW      ;RUNNING UNDER MEM MGMNT?
2297  010652' 001406                 BEQ    72$                 ;Y,N-72$
2298  010654' 032766 140000 000020  BIT    #140000,16.(SP)     ;WERE WE IN USER MODE?
2299  010662' 001402                 BEQ    72$                 ;Y,N-72$
2300  010664' 005726                 TST    (SP)+               ;REMOVE OTHER ADR
2301  010666' 006506                 MFPI   SP                  ;GET USER'S STK PNTR
2302  010670' 005046          72$:   CLR    -(SP)               ;SET PROCESSOR PRIORITY TO ZERO
2303  010672' 012746 010700'         MOV    #74$,-(SP)
2304  010676' 000002                 RTI
2305  010700' 032777 000100 000240 74$:  BIT    #100,@CTWDRA        ;CONS TERM WRITE INT ENABLE SET?
2306  010706' 001374                 BNE    74$                 ;(N,Y-74$)
2307  010710' 052777 100000 000012' BIS    #CTRQBP,@CSYSFW     ;SET CONS REQ BEING PROC FLG
2308  010716' 042714 001000          BIC    #CTCREQ,(R4)        ;RESET CONSOLE REQ FLAG
2309  010722' 000177 000014'         JMP    @USRINT             ;GO TO MPG USER INT ENTRY
2310
2311
2312
2313                                                ;CONSOLE TERMINAL READ TERMINATE S/R
2314
2315  010726' 032767 000400 000122 CTRTRM: BIT    #CTRDON,CTFLGW      ;IS THE READER DONE FLAG SET?
2316  010734' 001403                 BEQ    1$                  ;(Y,N-1$)
```

```
2317  010736' 042767  000441  000112          BIC     #CTRBSY+CTRDON+CTEBSY,CTFLGW    ;RESET READER FLAGS
2318  010744' 000205                  1$:     RTS     R5                              ;EXIT IN-LINE
2319
2320                                          ;CONSOLE TERMINAL PUT IN ECHO BUFFER S/R
2321
2322  010746' 012503                  CTRPEB: MOV     (R5)+,R3                        ;GET CHARACTERS TO STORE
2323  010750' 105714                          TSTB    (R4)                            ;IS ECHO FLAG SET?
2324  010752' 100023                          BPL     CTRPEX                          ;(Y,N-CTRPEX)
2325  010754' 010546                  CTWPEB: MOV     R5,-(SP)                        ;SAVE WORK REG
2326  010756' 012705  011104'                 MOV     #CTEIAD,R5                      ;SU ADR OF ECHO BUF CNTRL WDS
2327  010762' 021565  000006          70$:    CMP     (R5),6(R5)                      ;INPUT ADR AT END OF BUF?
2328  010766' 001002                          BNE     80$                             ;(Y,N-80$)
2329  010770' 162715  000024                  SUB     #24,(R5)                        ;SET UP BUFFER START ADR
2330  010774' 110335                  80$:    MOVB    R3,@(R5)+                       ;MOVE CHAR TO ECHO BUF
2331  010776' 005215                          INC     (R5)                            ;ADD 1 TO BUFFER COUNT
2332  011000' 005245                          INC     -(R5)                           ;ADD 1 TO BUFFER ADR
2333  011002' 105003                          CLRB    R3                              ;CLEAR CHAR JUST STORED
2334  011004' 000303                          SWAB    R3                              ;SET UP FOR NEXT CHAR
2335  011006' 005703                          TST     R3                              ;ANOTHER CHAR TO STORE?
2336  011010' 001364                          BNE     70$                             ;(N,Y-70$)
2337  011012' 012605                          MOV     (SP)+,R5                        ;RESTORE WORK REG
2338  011014' 052777  000100  000124          BIS     #100,@CTWDRA                    ;SET WRITE INT ENABLE
2339  011022' 000205                  CTRPEX: RTS     R5                              ;EXIT IN-LINE
2340
2341
2342
2343
2344                                          ;INTERRUPT SUBROUTINES
2345
2346
2347                                          ;SAVE REGISTERS R0 THRU R5
2348
2349  011024' 010446                  CSVREG: MOV     R4,-(SP)                        ;SAVE REGISTERS 0 THRU 4
2350  011026' 010346                          MOV     R3,-(SP)                        ;R5 IS SAVED BY JSR TO
2351  011030' 010246                          MOV     R2,-(SP)                        ;THIS SUBROUTINE
2352  011032' 010146                          MOV     R1,-(SP)
2353  011034' 010046                          MOV     R0,-(SP)
2354  011036' 010507                          MOV     R5,PC                           ;EXIT
2355
2356
2357                                          ;RESTORE REGISTERS R0 THRU R5
2358
2359  011040' 005726                  CRTREG: TST     (SP)+                           ;TAKE THIS JSR'S R5 OFF STACK
2360  011042' 012600                          MOV     (SP)+,R0                        ;RELOAD REG'S
2361  011044' 012601                          MOV     (SP)+,R1
2362  011046' 012602                          MOV     (SP)+,R2
2363  011050' 012603                          MOV     (SP)+,R3
2364  011052' 012604                          MOV     (SP)+,R4
2365  011054' 000205                          RTS     R5                              ;RESTORE OLD R5 & EXIT
```

```
                                        .SBTTL  CONSOLE TERMNAL INTERRUPT CONSTANTS

011056' 000000          CTFLGW: .WORD   0                       ;CONSOLE TERMINAL FLAGWORD

        000001                  CTRBSY= 1                       ;  READ BUSY
        000002                  CTWBSY= 2                       ;  WRITE BUSY
        000040                  CTEBSY= 40                      ;  ECHO BUSY
        000100                  CTRRUB= 100                     ;  READ RUBOUT STRING
        000200                  CTRDEC= 200                     ;  READ WITH ECHO
        000400                  CTRDON= 400                     ;  READ DONE
        001000                  CTCREQ= 1000                    ;  CONSOLE REQUEST
        002000                  CTWAIT= 2000                    ;  SOFTWARE "WAIT" FLAG
        100000                  CTWUSR= 100000                  ;  USER WRITE

                                ;CTWCNT MUST FOLLOW CTFLGW

011060' 000000          CTWCNT: .WORD   0                       ;CONSOLE TERMINAL WRITE COUNT
011062' 000000          CTWADR: .WORD   0                       ;CT WR DATA ADR
011064' 000000          CTWUFW: .WORD   0                       ;*;CT WRITE USER FLGWD ADR

011066' 000000          CTRCNT: .WORD   0                       ;CT RD DATA COUNT (CURRENT)
011070' 000000          CTRADR: .WORD   0                       ;CT RD DATA ADR (CURRENT)
011072' 000000                  .WORD   0                       ;CT RD DATA COUNT (ORIG)
011074' 000000                  .WORD   0                       ;CT RD DATA ADR (ORIG)

011076' 000000          CTFTCT: .WORD   0                       ;CT WR FILL/TAB COUNT
011100' 000000          CTFSCH: .WORD   0                       ;CT WR FILL/SPACE CHAR
011102' 000010                  .WORD   10                      ;CT WR CURR SOFTWARE TAB CNT

        000024                  CTEBFL= 24
011104' 011114'         CTEIAD: .WORD   CTEBUF                  ;ECHO BUFFER INPUT ADR
011106' 000000          CTECNT: .WORD   0                       ;ECHO BUF COUNT
011110' 011114'                 .WORD   CTEBUF                  ;ECHO BUF OUTPUT ADR
011112' 011140'                 .WORD   CTEBUF+CTEBFL           ;ECHO BUF END ADR
011114' 000024          CTEBUF: .BLKB   CTEBFL                  ;ECHO BUFFER

                                ;CONSOLE TERMINAL CONSTANTS

011140' 177560          CTRDRA: .WORD   177560                  ;CONSOLE TERMINAL RD DEV REG ADR
011142' 000060          CTRVEC: .WORD   60                      ;CT RD INT VECTOR ADR
011144' 000200          CTRBR:  .WORD   200                     ;CT RD PRIORITY LEVEL (BR4)
011146' 177564          CTWDRA: .WORD   177564                  ;CT WR DEV REG ADR
011150' 000064          CTWVEC: .WORD   64                      ;CT WR INT VECTOR ADR
011152' 000200          CTWBR:  .WORD   200                     ;CT WR PRIORITY LEVEL (BR4)

011154' 000002          CTFCNT: .WORD   2                       ;CT WR FILL COUNT
011156' 000000          CTFCAR: .WORD   0                       ;CT WR FILL CHAR
011160' 000012          CTFID:  .WORD   012                     ;CT WR FILL I.D. CHAR
```

```
                                .SBTTL  PRINTER (LP11,LS11,LV11) INTERRUPT ROUTINE

                        ;*****************************************************************
                        ;
                        ;  THIS ROUTINE PROCESSES INTERRUPTS FOR THE LP11, LS11, AND LV11
                        ;  PRINTERS. AN ERROR CONDITION WILL CAUSE IMMEDIATE TERMINATION
                        ;  OF THE I/O OPERATION. MAXIMUM LINE LENGTH WILL BE
                        ;  80 CHARACTERS. ANY ABOVE THAT WILL BE TRUNCATED. TERMINATION
                        ;  INCLUDES THE RESETTING OF USER 'W4IOT' & 'CTPRIO' FLAGS
                        ;  IF A USER I/O.
                        ;
                        ;*****************************************************************

011162' 004567 177636   CPRINT: JSR     R5,CSVREG               ;GO SAVE ALL REGISTERS
011166' 012701 011432'          MOV     #CPRCNT,R1              ;SET UP INFO STORAGE AREA ADR
011172' 016700 000250           MOV     PRTDRA,R0               ;GET DEV REG ADR
011176' 005710                  TST     (R0)                    ;PRINTER ERROR?
011200' 100457                  BMI     CPRTRM                  ;N,Y-CPRTRM
011202' 106327                  ASLB    (PC)+                   ;DOING A TAB?
011204' 000000          CPRTFG: .WORD   0
011206' 001051                  BNE     CPRDOT                  ;N,Y-CPRDOT
011210' 005711          CPRIGN: TST     (R1)                    ;CHAR CNT = 0?
011212' 001452                  BEQ     CPRTRM                  ;N,Y-CPRTRM
011214' 005311                  DEC     (R1)                    ;DECR CNT BY 1
011216' 115102                  MOVB    @-(R1),R2               ;GET DATA CHAR
011220' 005221                  INC     (R1)+                   ;INCR DATA ADR BY 1
011222' 120227 000040           CMPB    R2,#40                  ;THIS A CONTROL CHAR?
011226' 103413                  BLO     CPRCTC                  ;N,Y-CPRCTC
011230' 005327          CPRMCK: DEC     (PC)+                   ;DECR LINE CHAR COUNT
011232' 000120          CPRLNC: .WORD   80.
011234' 002765                  BLT     CPRIGN                  ;ROOM LEFT? (Y,N-CPRIGN)
011236' 106327                  ASLB    (PC)+                   ;UPDATE RUNNING TAB COUNT
011240' 000001          CPRTBC: .WORD   1
011242' 001424                  BEQ     CPRRTC                  ;CNT EXHAUSTED? (N,Y-CPRRTC)
011244' 110260 000002   CPRSCH: MOVB    R2,2(R0)                ;SEND CHAR TO PRINTER
011250' 004567 177564   CPREX:  JSR     R5,CRTREG               ;RESTORE REGISTERS
011254' 000002                  RTI                             ;RETURN AFTER INTERRUPT

011256' 120227 000011   CPRCTC: CMPB    R2,#011                 ;CHAR A TAB?
011262' 001420                  BEQ     CPRTAB                  ;N,Y-CPRTAB
011264' 120227 000015           CMPB    R2,#015                 ;IS IT A CARR RETURN?
011270' 001406                  BEQ     CPRSTC                  ;N,Y-CPRSTC
011272' 120227 000012           CMPB    R2,#012                 ;A LINE FEED?
011276' 001403                  BEQ     CPRSTC                  ;N,Y-CPRSTC
011300' 120227 000014           CMPB    R2,#014                 ;A FORM FEED?
011304' 001341                  BNE     CPRIGN                  ;Y,N-CPRIGN -- IGNORE IT
011306' 012767 000120 177716    CPRSTC: MOV     #80.,CPRLNC     ;INITIALIZE LINE CHAR COUNT
011314' 012767 000001 177716    CPRRTC: MOV     #1,CPRTBC       ;INITIALIZE TAB COUNT
011322' 000750                  BR      CPRSCH                  ;GO SEND THE CHAR
011324' 016767 177710 177652    CPRTAB: MOV     CPRTBC,CPRTFG   ;SET UP CURR TAB COUNT
011332' 012702 000040           CPRDOT: MOV     #40,R2          ;SET UP SPACE CHAR
011336' 000734                  BR      CPRMCK                  ;GO CHECK FOR ROOM

011340' 005767 000062   CPRTRM: TST     CPRFWD                  ;THIS A USER PRINT?
011344' 100017                  BPL     10$                     ;Y,N-10$
011346' 004577 000002'          JSR     R5,@UFWCLR              ;RESET USER'S WAIT FOR
```

```
2477  011352' 000020                          .WORD   CTPRIO                  ;CONSOLE/PRINTER I/O FLAG
2478  011354' 000060                          .WORD   CPRUFW-.
2479  011356' 032777  040000  000012'         BIT     #CRUDED,@CSYSFW         ;SUPPOSED TO RESET USER DEDICATED?
2480  011364' 001731                          BEQ     CPREX                   ;Y,N-CPREX
2481  011366' 042777  064000  000012'         BIC     #CRUDED+CTUDED+CHPINT,@CSYSFW ;RESET SYS'S DED. FLAGS
2482  011374' 004577  000002'                 JSR     R5,@UFWCLR              ;RESET USER DEDICATED FLAG
2483  011400' 000040                          .WORD   SETDED
2484  011402' 000032                          .WORD   CPRUFW-.
2485  011404' 032777  004000  000012'  10$:   BIT     #CHPINT,@CSYSFW         ;INT ENB BEING HELD SET?
2486  011412' 001001                          BNE     20$                     ;N,Y-20$
2487  011414' 005010                          CLR     (R0)                    ;RESET PRINTER'S INT ENABLE
2488  011416' 042767  000001  000002  20$:   BIC     #CPRBSY,CPRFWD          ;RESET PRINTER'S BUSY FLAG
2489  011424' 000711                          BR      CPREX                   ;GO TO EXIT
2490
2491
2492
2493  011426' 000000                  CPRFWD: .WORD   0                       ;PRINTER FLAG WORD
2494          000001                          CPRBSY= 1                       ;  PRINTER IS BUSY
2495          000200                          CPRERR= 200                     ;  PRINT ERROR MSG ISSUED
2496          100000                          CPRUSR= 100000                  ;  USER PRINT FUNCTION
2497
2498  011430' 000000                  CPRADR: .WORD   0                       ;CURRENT DATA ADR
2499  011432' 000000                  CPRCNT: .WORD   0                       ;CURRENT CHAR COUNT
2500  011434' 000000                  CPRUFW: .WORD   0                       ;*;USER FLAGWORD ADR
2501
2502  011436'                         CLISTI:                                 ;LIST DEVICE INFO STORAGE
2503  011436' 000000                          .WORD   0                       ;  DEFAULT MDL CODE WORD
2504  011440' 000000                          .WORD   0                       ;  DEFAULT UNIT #
2505  011442' 000000                          .WORD   0                       ;  CURRENT MODEL CODE WORD
2506  011444' 000000                          .WORD   0                       ;  CURRENT UNIT #
2507  011446' 177514                  PRTDRA: .WORD   177514                  ;  DEV REG ADR
2508  011450' 000200                  PRTIVA: .WORD   200                     ;  INT VECTOR ADR
2509  011452' 000200                  PRTPSW: .WORD   200                     ;  PS WORD
2510  011454' 000000                          .WORD   0
2511
2512  011456' 000000                  UFWHLD: .WORD   0                       ;*;USER FLGWD ADR HOLD
```

```
                                        .SBTTL  PAPER TAPE HANDLER PARAMETER TABLE & ROUTINES

011460' 011532'                 PTPARM: .WORD   PTBOOT                   ;PAPER TAPE VECTOR TABLE
011462' 011532'                         .WORD   PTDIR
011464' 011532'                         .WORD   PTZERO
011466' 011532'                         .WORD   PTDELT
011470' 011556'                         .WORD   PTCLOS
011472' 011550'                         .WORD   PTENT
011474' 011536'                         .WORD   PTLOOK
011476' 011550'                         .WORD   PTALLC
011500' 011604'                         .WORD   PTDVR

011502' 000000                  PTUNIT: .WORD   0                        ;UNIT #
011504' 000000                  PTCREG: .WORD   0                        ;PC11 CMND REG ADR
011506' 000776                  PTWCNT: .WORD   510.                     ;BYTE COUNT
011510' 001146'                 PTBADR: .WORD   BUF+2                    ;BUS ADR
011512' 000000                  PTBLK:  .WORD   0                        ;BLOCK #
011514' 000000                  PTCMND: .WORD   0                        ;COMMAND
011516' 111221                  PTRD:   .WORD   111221                   ;READ COMMAND--MOVB (R2),(R1)+
011520' 112112                  PTWR:   .WORD   112112                   ;WRITE COMMAND--MOVB (R1)+,(R2)
011522' 000000                  PTRBCT: .WORD   0                        ;REQUESTED BLOCK COUNT
011524' 011530'                 PTDPTR: .WORD   PTDBLK                   ;POINTS TO 1ST DIR BLOCK
011526' 000000                  PTLBLK: .WORD   0                        ;LAST BLOCK # ALLOCATED
011530'                         PTPEND:
011530' 000000                  PTDBLK: .WORD   0                        ;0 INDICATES NON-DIRECTORY DEVICE

                                        ;PAPER TAPE ROUTINES

011532'                         PTBOOT:
011532'                         PTDIR:
011532'                         PTZERO:
011532' 000167  174660          PTDELT: JMP     INVCMD                   ;ISSUE INVALID CMND ERROR MSG
011536' 012765  000001  177744  PTLOOK: MOV     #1,X1STBK(R5)            ;SET UP DUMMY BLK # IN DDB
011544' 062716  000002                  ADD     #2,(SP)                  ;SET UP SUCCESSFUL EXIT
011550'                         PTENT:
011550' 012765  000001  177750  PTALLC: MOV     #1,XLSTBK(R5)            ;FAKE A BLOCK #
011556' 000207                  PTCLOS: RTS     PC                       ;EXIT IN-LINE

                                        ;DEV ADDRESS  SET UP  ROUTINES

011560' 012767  177550  177716  SETPR:  MOV     #177550,PTCREG           ;SET UP PTR DEV REG ADR
011566' 000403                          BR      PTCOM                    ;GO TO COMMON POINT
011570' 012767  177554  177706  SETPP:  MOV     #177554,PTCREG           ;SET  UP PTP DEV REG ADR
011576' 012700  011460'         PTCOM:  MOV     #PTPARM,R0               ;SET UP PT PARAM TBL ADR
011602' 000207                          RTS     PC                       ;EXIT IN-LINE
```

```
                                                .SBTTL  PAPER TAPE HANDLER DRIVER

011604' 005027                          PTDVR:  CLR     (PC)+           ;CLEAR DATA XFER FLAG
011606' 000000                          PTNOUT: .WORD   0               ;DATA TRANSFER FLAG
011610' 016503 000000                           MOV     XCM(R5),R3      ;GET STATUS REG ADR
011614' 010302                                  MOV     R3,R2           ;SET UP DATA REG ADR
011616' 005722                                  TST     (R2)+           ;ERROR BIT SET?
011620' 100470                                  BMI     PTERR           ;(N,Y-PTERR)
011622' 012701 001146'                          MOV     #BUF+2,R1       ;SET UP BUFFER ADR
011626' 012700 000776                           MOV     #510.,R0        ;SET UP BYTE COUNT
011632' 016567 000010 000006                    MOV     XCO(R5),PT2     ;SET UP MOVB INST
011640' 005213                          PT1:    INC     (R3)            ;SET THE "GO" BIT IF THERE
011642' 004767 000062                           JSR     PC,PTIMER       ;GO WAIT FOR DONE
011646' 000240                          PT2:    NOP                     ;CHANGED TO EITHER 111221 OR 112112
011650' 001014                                  BNE     PT3             ;BR IF DATA IS NON-ZERO
011652' 005767 177730                           TST     PTNOUT          ;ANY DATA TRANSFERRED?
011656' 001011                                  BNE     PT3             ;BR IF YES
011660' 005765 177716                           TST     XWCTR(R5)       ;NO, THIS 1ST BLK?
011664' 001006                                  BNE     PT3             ;BR IF NOT
011666' 026567 000010 177622                    CMP     XCO(R5),PTRD    ;THIS A READ OPERATION?
011674' 001002                                  BNE     PT3             ;BR IF NOT
011676' 005301                                  DEC     R1              ;BKUP XFER ADR, IGNORE LEADING 0'S
011700' 000757                                  BR      PT1             ;GO KEEP I/O GOING
011702' 005267 177700                   PT3:    INC     PTNOUT          ;SET DATA XFERRED FLAG
011706' 001405                                  BEQ     PTDONE
011710' 005300                                  DEC     R0              ;DECR BYTE COUNT
011712' 003352                                  BGT     PT1             ;CNT = 0? (Y,N-PT1)
011714' 012767 000001 167222                    MOV     #1,BUF          ;SET UP BLOCK #
011722' 005265 177716                   PTDONE: INC     XWCTR(R5)       ;UP BLKS PROCESSED CNT
011726' 000207                                  RTS     PC              ;EXIT IN-LINE

011730' 005027                          PTIMER: CLR     (PC)+           ;RESET DELAY COUNTER
011732' 000000                          PTCNT:  .WORD   0               ;DELAY COUNTER
011734' 032713 100200                   PTIME1: BIT     #100200,(R3)    ;TEST FOR ERROR OR DONE
011740' 100415                                  BMI     PTOUT           ;ERROR SET? (N,Y-PTOUT)
011742' 003016                                  BGT     PTIMEX          ;DONE SET? (N,Y-PTIMEX)
011744' 026767 177762 177760                    CMP     PTCNT,PTCNT     ;DELAY A BIT
011752' 026767 177754 177752                    CMP     PTCNT,PTCNT
011760' 005267 177746                           INC     PTCNT           ;INCR DELAY COUNTER
011764' 001363                                  BNE     PTIME1          ;TIMED OUT? (Y,N-PTIME1)
011766' 005767 177614                           TST     PTNOUT          ;DONE ANY OUTPUT?
011772' 001403                                  BEQ     PTERR           ;Y,N-PTERR
011774' 005067 167144                   PTOUT:  CLR     BUF             ;EOF ON THE DEVICE
012000' 000207                          PTIMEX: RTS     PC              ;EXIT IN-LINE FROM S/R

012002' 000167 174316                   PTERR:  JMP     DEVERR          ;GO REPORT DEVICE ERROR
                  001                           .IF DF MIMIC



                                        ALPT1=  .-LOC600
                                        ALFCT1= ALPT1&000077+8./2
                  002                           .IF NE ALFCT1
                                        ALQUN1= 31.-ALFCT1
                                                .BLKW   ALQUN1          ;ALIGN TO NEXT 32 WORD BNDRY - 8
                                                .WORD   0
```

```
2619            001                         .ENDC
2620            000                         .ENDC
```

```
                                        .SBTTL  LOAD MEDIA HANDLER


012006' 007016          EMDLTB: .WORD   7016                    ;PAPER TAPE (PC11) MDL CODE
012010' 011560'         PTSUAD: .WORD   SETPR                   ;PAPER TAPE RDR (PC11) SU ADR
012012' 007014                  .WORD   7014                    ;PAPER TAPE RDR (PR11) MDL CODE
012014' 011560'                 .WORD   SETPR                   ;SU ADR

                                ;THE REST OF THIS TABLE IS CONTAINED IN
                                ;THE FIRST PART OF THE LOAD MEDIA HANDLER.


000000'                         .CSECT  HANDLR


                        ;  THE LOAD MEDIA DEVICE HANDLER WILL RESIDE IN THIS
                        ;  AREA OF MEMORY AND WILL BE FOLLOWED IN MEMORY
                        ;  BY THE ONE TIME HOUSEKEEPING ROUTINE.


                        ;  THE LISTINGS THAT CONTAIN THE VARIOUS HANDLERS WHICH
                        ;  OCCUPY THIS AREA ARE:
                        ;
                        ;    TCMPG  = DTE0A-? - TC11 HANDLER
                        ;    RKMPG  = DTE1A-? - RK11 HANDLER
                        ;    TMMPG  = DTE2A-? - TM11 HANDLER
                        ;    THMPG  = DTE3A-? - TM02 HANDLER
                        ;    RXMPG  = DTE4A-? - RX11 HANDLER
                        ;    RBMPG  = DTE5A-? - RP04 HANDLER
```

```
                                        .SBTTL  MPG EXEC TEMPORARY ONE TIME HOUSEKEEPING

        000000'                         .CSECT  EXHSKP


000000'                         HKLOCZ:
            001                         .IF NDF MIMIC

000000' 000000                  UFWSET: .WORD   XXXX
000002' 000000                  UFWCLR: .WORD   XXXX
000004' 000000                  UFWTST: .WORD   XXXX
000006' 000000                  BINASC: .WORD   XXXX
000010' 000000                  BTASLB: .WORD   XXXX
000012' 000014'                 CSYSFW: .WORD   USRINT
000014' 000000                  USRINT: .WORD   XXXX

        000016'                 MPGBGN  =.              ;MPG WILL BE LOADED STARTING HERE OR THE
                                                        ;NEXT 32 WORD BOUNDARY IF USING MEM MGMNT.
            000                         .ENDC

                                        ;THE FOLLOWING CODE IS EXECUTED ONE TIME
                                        ;WHEN THE MPG EXEC IS FIRST LOADED. THIS CODE
                                        ;WILL BE DESTROYED WHEN THE MPG PROGRAM
                                        ;IS LOADED INTO THIS AREA.


                                        ;FORMAT VECTOR LOCATIONS 0 THRU 276.
                                        ;SET THE PSW TO 0.
                                        ;TEST IF CONSOLE TERMINAL IS PRESENT.
                                        ;INITIALIZE CONSOLE TERMINAL ADDRESSES.

000016' 005000                  E1THKP: CLR     R0                      ;SET LOCATIONS 0 & 2
000020' 005020                          CLR     (R0)+                   ;TO HALT'S
000022' 005020                          CLR     (R0)+
000024' 010001                          MOV     R0,R1                   ;SET LOCATIONS 4 THRU 276 TO
000026' 005721                          TST     (R1)+                   ;.+2 AND HALT.
000030' 010120                  10$:    MOV     R1,(R0)+                ;DO NOT CHANGE LOCATIONS
000032' 005020                          CLR     (R0)+                   ;40 & 42
000034' 062701  000004          20$:    ADD     #4,R1
000040' 020027  000300          30$:    CMP     R0,#300
000044' 001406                          BEQ     40$
000046' 020027  000040                  CMP     R0,#40
000052' 001366                          BNE     10$
000054' 062700  000004                  ADD     #4,R0
000060' 000765                          BR      20$
000062' 012737  000002  000102  40$:    MOV     #2,@#102                ;INIT LSI-11 CLK VECTOR
000070' 005046                          CLR     -(SP)                   ;SET UP PSW OF 0
000072' 012746  000100'                 MOV     #TTYTST,-(SP)           ;SET UP ADR OF INST AFTER RTI
000076' 000002                          RTI                             ;POP PSW & PC OFF STACK
000100' 012737  000124' 000004  TTYTST: MOV     #TTYTRP,@#4             ;SET UP BUS TIMEOUT TRAP
000106' 105777  011146'                 TSTB    @CTWDRA                 ;IS A TERMINAL THERE FOR WRITE?
000112' 100407                          BMI     FMTVCT                  ;N,Y-FMTVCT
000114' 012700  011140'         TTYERR: MOV     #CTRDRA,R0              ;SET CONS TERM INFO ADR IN R0
000120' 000000                          HALT                            ;STOP; CONSOLE TERM NOT THERE.
                                                                        ;ALLOW USER TO CHANGE CONS
```

```
                                                                        ;TERM PRESET INFO
000122' 000766                          BR      TTYTST                  ;GO TRY WITH NEW DATA
000124' 012716  000114'         TTYTRP: MOV     #TTYERR,(SP)            ;SET UP ERR RET ADR
000130' 000002                          RTI                             ;EXIT FROM TRAP
000132' 012700  011142'         FMTVCT: MOV     #CTRVEC,R0              ;SET UP CONS TERM RD & WR
000136' 012001                          MOV     (R0)+,R1                ;INT VECTOR LOCATIONS
000140' 012721  010162'                 MOV     #CTRINT,(R1)+           ;STORE RD INT ROUT ADR
000144' 012021                          MOV     (R0)+,(R1)+             ;STORE RD PS WORD
000146' 005720                          TST     (R0)+                   ;BY PASS WR DEV REG ADR
000150' 012001                          MOV     (R0)+,R1                ;GET WR INT VECT ADR
000152' 012721  007626'                 MOV     #CTWINT,(R1)+           ;STORE WRITE INT ROUT ADR
000156' 012021                          MOV     (R0)+,(R1)+             ;STORE WR PS WORD


                                        ;ISSUE EXECUTIVE'S TITLE MESSAGE

000160' 004567  005654'                 JSR     R5,MES                  ;ISSUE PORTION OF MSG THAT
000164' 000000G                         .WORD   EXCID                   ;IS IN THE LOADER OR HANDLER
000166' 004567  005654'                 JSR     R5,MES                  ;ISSUE TRAILING PORTION
000172' 002040'                         .WORD   EXCIDT                  ;OF THE MSG


                                        ;CLEAR MEM & DETERMINE MEM SIZE

000174' 012737  000236' 000004  EHKMEM: MOV     #EHKBAT,@#4             ;SET BUS ADR TRAP ROUT ADR
000202' 016700  001136                  MOV     LIMIT+2,R0              ;SET UP END OF EXEC ADR
000206' 005010                  10$:    CLR     (R0)                    ;CLEAR WORD OF MEMORY
000210' 005720                          TST     (R0)+                   ;POINT TO NEXT LOCATION
000212' 020027  160000                  CMP     R0,#160000              ;REACHED END OF MAX MEM?
000216' 001373                          BNE     10$                     ;Y,N-10$
000220' 020027  100000          EMFND:  CMP     R0,#100000              ;MEM 16K WORDS OR LARGER?
000224' 103407                          BLO     EINSFM                  ;Y,N-EINSFM
000226' 005740                          TST     -(R0)                   ;SET UP LAST WORD ADR
000230' 010067  000466'                 MOV     R0,ENDMEM               ;SAVE IT
000234' 000410                          BR      EGTDAT                  ;GO TO GET DATE ROUTINE

000236' 012716  000220'         EHKBAT: MOV     #EMFND,(SP)             ;CHANGE RETURN ADR ON STACK
000242' 000002                          RTI                             ;RETURN AFTER OUT OF MEM TRAP

000244' 004567  005654'         EINSFM: JSR     R5,MES                  ;ISSUE INSUFFICIENT MEMORY
000250' 000136'                         .WORD   ISUFM                   ;ERROR MSG
000252' 000000                          HALT                            ;INSUF MEM HALT
000254' 000747                          BR      EHKMEM                  ;GO CK MEM AGAIN


                                        ;GET TODAY'S DATE & PACK INTO DOS COMPATIBLE FORMAT

000256' 004567  005654'         EGTDAT: JSR     R5,MES                  ;ISSUE "ENTER DATE" MSG
000262' 001454'                         .WORD   EDMSG
000264' 004567  007124'                 JSR     R5,CTRD                 ;ISSUE READ FOR THE REPLY
000270' 001024                          .WORD   EKYBUF-.
000272' 000025                          .WORD   21.
000274' 012767  001314' 001010          MOV     #EKYBUF,EKBPTR          ;INITIALIZE REPLY DATA POINTER
000302' 004767  001144'                 JSR     PC,ATOI                 ;GO SCAN FOR DAY VALUE
000306' 005702                          TST     R2                      ;IS IT 0?
000310' 001527                          BEQ     EDATER                  ;N,Y-EDATER
```

```
2764  000312' 010204                        MOV     R2,R4           ;SAVE DAY VALUE
2765  000314' 010267  000072                MOV     R2,33$
2766  000320' 004767  000212                JSR     PC,EKBDSK       ;GO CHECK FOR DATA TO BE SKIPPED
2767  000324' 012701  006214'               MOV     #EMONTB,R1      ;SET UP MONTH TABLE START ADR
2768  000330' 012702  000002        10$:    MOV     #2,R2           ;SET UP COMPARE CNT -1
2769  000334' 122021                20$:    CMPB    (R0)+,(R1)+     ;MONTH CHAR'S COMPARE?
2770  000336' 001413                        BEQ     30$             ;N,Y-30$
2771  000340' 060201                        ADD     R2,R1           ;INCR PAST REST OF CHAR'S
2772  000342' 112103                        MOVB    (R1)+,R3        ;GET THIS MONTH'S # OF DAYS
2773  000344' 060304                        ADD     R3,R4           ;ADD IT INTO RUNNING TOTAL
2774  000346' 020127  006274'               CMP     R1,#EMONTB+48.  ;CHECKED ALL MONTHS?
2775  000352' 001506                        BEQ     EDATER          ;N,Y-EDATER
2776  000354' 012703  000003                MOV     #3,R3           ;BACKUP KYBD DATA PNTR
2777  000360' 160203                        SUB     R2,R3
2778  000362' 160300                        SUB     R3,R0
2779  000364' 000761                        BR      10$             ;GO CK NXT MONTH
2780  000366' 005302                30$:    DEC     R2              ;DECR COMPARE CNT
2781  000370' 100361                        BPL     20$             ;COMPARED 3 CHAR'S? (Y,N-20$)
2782  000372' 010067  000714                MOV     R0,EKBPTR       ;STORE NEW DATA PNTR
2783  000376' 004767  000134                JSR     PC,EKBDSK       ;DO SELECTED DATA SKIP
2784  000402' 004767  001144'               JSR     PC,ATOI         ;SCAN & CONVERT YEAR VALUE
2785  000406' 111103                        MOVB    (R1),R3         ;GET # OF DAYS IN THIS MONTH
2786  000410' 022703                        CMP     (PC)+,R3        ;# OF DAYS ENTERED TOO LARGE?
2787  000412' 000000                33$:    .WORD   XXXX
2788  000414' 101412                        BLOS    36$             ;Y,N-36$
2789  000416' 032702  000003                BIT     #3,R2           ;THIS A LEAP YEAR?
2790  000422' 001062                        BNE     EDATER          ;Y,N-EDATER
2791  000424' 020127  006223'               CMP     R1,#EMONTB+7    ;IS THIS FEBRUARY?
2792  000430' 001057                        BNE     EDATER          ;Y,N-EDATER
2793  000432' 026727  177754  000035        CMP     33$,#29.        ;IS DAY = 29?
2794  000440' 001053                        BNE     EDATER          ;Y,N-EDATER
2795  000442' 032702  000003        36$:    BIT     #3,R2           ;THIS A LEAP YEAR?
2796  000446' 001004                        BNE     40$             ;Y,N-40$
2797  000450' 020127  006224'               CMP     R1,#EMONTB+8.   ;IS MONTH MARCH OR LATER?
2798  000454' 103401                        BLO     40$             ;Y,N-40$
2799  000456' 005204                        INC     R4              ;ADD TO RUN TOT FOR FEB 29
2800  000460' 020227  000143        40$:    CMP     R2,#99.         ;YEAR ENTERED > 99?
2801  000464' 101041                        BHI     EDATER          ;N,Y-EDATER
2802  000466' 162702  000106                SUB     #70.,R2         ;SUB BASE VALUE FROM YEAR # ENTERED
2803  000472' 100436                        BMI     EDATER          ;LESS THAN BASE? (N,Y-DATER)
2804  000474' 001404                50$:    BEQ     60$             ;AT BASE YEAR? (N,Y-60$)
2805  000476' 062704  001750                ADD     #1000.,R4       ;ADD YEAR INCR VALUE TO RUN TOT
2806  000502' 005302                        DEC     R2              ;DECR YEAR VALUE
2807  000504' 000773                        BR      50$             ;GO CK FOR BASE
2808  000506' 010467  000470'       60$:    MOV     R4,TODAY        ;STORE RUN TOT AS M/D/Y CODE
2809  000512' 004767  000020                JSR     PC,EKBDSK       ;DO SELECTED DATA SKIP
2810  000516' 005067  000336                CLR     DATELF          ;RESET LF TERMINATOR FLAG
2811  000522' 121027  000012                CMPB    (R0),#012       ;LF TERMINATE DATE REPLY?
2812  000526' 001002                        BNE     70$             ;Y,N-70$
2813  000530' 005267  000324                INC     DATELF          ;SET LF TERMINATOR REPLY FLG
2814  000534' 000421                70$:    BR      ESYDSU          ;GO SET UP FOR LOAD DEV
2815
2816  000536' 016700  000550        EKBDSK: MOV     EKBPTR,R0       ;GET CURR DATA PNTR
2817  000542' 121027  000040        100$:   CMPB    (R0),#40        ;DATA CHAR A SPACE?
2818  000546' 001403                        BEQ     110$            ;N,Y-110$
2819  000550' 121027  000055                CMPB    (R0),#'-        ;A MINUS SIGN?
```

```
                                        BNE     120$                    ;Y,N-120$
                                110$:   INC     R0                      ;ADD 1 TO DATA ADR
                                        BR      100$                    ;GO CK NXT CHAR
                                120$:   MOV     R0,EKBPTR               ;STORE NEW PNTR
                                        RTS     PC                      ;EXIT IN-LINE

                                EDATER: JSR     R5,MES                  ;ISSUE 'INV DATE' MSG
                                        .WORD   EDERMG
                                        BR      EGTDAT                  ;GO TRY AGAIN


                                        ;INITIALIZE FOR SYSTEM LOAD DEVICE

                                ESYDSU: CLR     R3                      ;INITIALIZE FOR NO LINE FEED
                                        MOV     (PC)+,R0                ;GET LOAD DEV UNIT #
                                LDUNUM: .WORD   -1                      ;LOAD DEV UNIT # STORAGE
                                        BPL     70$                     ;IS THERE ONE? (N,Y-70$)
                                60$:    JSR     R5,MES                  ;ISSUE MSG TO REQUEST ONE
                                        .WORD   LDUREQ
                                        JSR     PC,OATOB                ;GET REPLY & CONVERT TO BINARY
                                        BR      60$                     ;DATA ERROR? (N,Y-60$)
                                        TST     R0                      ;ANY DATA RECEIVED?
                                        BEQ     70$                     ;Y,N-70$
                                        MOV     R1,R0                   ;MOVE DATA RC'VD TO CORRECT REG
                                        CMP     R0,#8.                  ;UNIT # TOO LARGE?
                                        BLO     70$                     ;Y,N-70$
                                        JSR     R5,MES                  ;ISSUE ERROR MSG
                                        .WORD   INVDMG
                                        BR      60$                     ;GO ASK AGAIN
                                70$:    MOV     R0,CFTCHI+2             ;STORE AS LOAD DEV DEFAULT UNIT #
                                        MOV     R0,CSAVEI+2
                                        MOV     R0,CFTCHI+6             ;STORE AS CURRENT UNIT #
                                        MOV     R0,CSAVEI+6
                                        CMPB    #012,R3                 ;LINE FEED TERMINATOR?
                                        BNE     CK4MM                   ;Y,N-CK4MM
                                        INC     DATELF                  ;SET LF TERM FLG


                                        ;DETERMINE IF MEMORY MANAGEMENT IS ON THE SYSTEM AND
                                        ;IF IT IS, ASK IF THEY WANT TO USE IT.
                                        ;IF USING IT, ALIGN MPG START ADR TO A 32 WORD BOUNDARY.

                                CK4MM:  MOV     #MMTRAP,@#4             ;SET UP BUS ADR TRAP ROUT ADR
                                        TST     @#MMMR0                 ;TEST IF MMMR0 REG IS THERE

                                        ;IF NO TRAP, WE DO THE FOLLOWING

                                80$:    JSR     R5,MES                  ;ASK IF THEY WANT TO USE MEM
                                        .WORD   USEMM                   ;MGMNT
                                        JSR     R5,CTRD                 ;ISSUE READ FOR THE REPLY
                                        .WORD   EKYBUF-.
                                        .WORD   21.
                                        JSR     PC,CKRPLY               ;CHECK FOR A REPLY
                                        BR      80$                     ;BR IF INV DATA
                                        BR      85$                     ;BR IF Y, <CR>, OR <LF>
                                        BR      MMCKRT                  ;BR IF N
```

```
                    001                             .IF DF MIMIC
                                                    .IFT
                                        85$:        MOV     #MPGBGN,R0          ;GET CURRENT MPG START ADR
                                                    .IFF
000744' 012767  040515  000034'         85$:        MOV     #"MA,MPGID          ;CHANGE MPG'S FILENAME
000752' 012700  000016'                             MOV     #MPGBGN,R0          ;GET CURRENT MPG START ADR
                    000                             .ENDC
                    001                             .IF DF DEBUG
                                                    .IFF
000756' 032700  000077                              BIT     #77,R0              ;IS IT ON A 32 WD BNDRY?
                                                    .IFT
                                                    BIT     #7777,R0            ;IS IT ON A 2048 WD BNDRY?
                    000                             .ENDC
000762' 001414                                      BEQ     MMCKRT              ;N,Y-MMCKRT
000764' 010001                                      MOV     R0,R1               ;SAVE CURR ADR
                    001                             .IF DF DEBUG
                                                    .IFF
000766' 042700  000077                              BIC     #77,R0              ;ALIGN ADR TO NXT 32 WD BNDRY
000772' 062700  000100                              ADD     #100,R0
                                                    .IFT
                                                    BIC     #7777,R0            ;ALIGN ADR TO NEXT 2048 WD BNDRY
                                                    ADD     #10000,R0
                    000                             .ENDC
000776' 160100                                      SUB     R1,R0               ;GET POSITION ADJ FACTOR
001000' 010067  000120'                             MOV     R0,POSADJ           ;STORE IT FOR MPG'S 1 TIME HSKP
001004' 060067  000472'                             ADD     R0,MPGST            ;UPDATE MPG'S START ADR
                    001                             .IF NDF MIMIC
001010' 060067  000046'                             ADD     R0,MPGADR           ;UPDATE MPG'S LOAD ADR
                    000                             .ENDC
001014' 120327  000040          MMCKRT:     CMPB    R3,#040             ;REPLY BYTE A LINE TERMINATOR?
001020' 103402                              BLO     90$                 ;N,Y-90$
001022' 112203                              MOVB    (R2)+,R3            ;GET NEXT REPLY BYTE
001024' 000773                              BR      MMCKRT              ;GO CK IT
001026' 120327  000012          90$:        CMPB    R3,#012             ;IS IT A LINE FEED?
001032' 001002                              BNE     MMCKEX              ;Y,N-MMCKEX
001034' 005267  000020                      INC     DATELF              ;SET LF TERM FLAG
001040' 012737  000006  000004  MMCKEX:     MOV     #6,@#4              ;RESTORE TRAP VECTOR
001046' 000403                              BR      CTCONS              ;GO TO NEXT HSKP FUNCTION

001050' 012716  001040'         MMTRAP:     MOV     #MMCKEX,(SP)        ;BYPASS REST OF THIS FUNCTION
001054' 000002                              RTI                         ;RETURN AFTER THIS TRAP


                                ;REQUEST CONSOLE TERMINAL CONSTANTS CHANGE

001056' 005727                  CTCONS:     TST     (PC)+               ;LF TERMINATOR ON DATE REPLY?
001060' 000000                  DATELF:     .WORD   0                   ;DATE LF TERMINATOR FLAG
001062' 001072                              BNE     RELGBL              ;N,Y-RELGBL
001064' 004567  005654'         5$:         JSR     R5,MES              ;ISSUE TITLE & FIRST REQ MSG
001070' 001522'                             .WORD   CON1MG
001072' 004767  001240'                     JSR     PC,OATOB            ;GET REPLY & CONVERT TO BINARY
001076' 000772                              BR      5$                  ;ERROR?  (N,Y-5$)
001100' 005700                              TST     R0                  ;DATA RECEIVED?
001102' 001402                              BEQ     10$                 ;Y,N-10$
001104' 010167  011160'                     MOV     R1,CTFID            ;STORE NEW I.D. CHAR
001110' 122703  000012          10$:        CMPB    #012,R3             ;LINE FEED TERMINATOR?
```

```
2932  001114' 001455                    BEQ     RELGBL                  ;N,Y-RELGBL
2933  001116' 004567  005654'  15$:     JSR     R5,MES                  ;ISSUE SECOND MSG
2934  001122' 001605'                   .WORD   CON2MG
2935  001124' 004767  001240'           JSR     PC,OATOB                ;GET REPLY & CONVERT
2936  001130' 000772                    BR      15$                     ;ERROR?  (N,Y-15$)
2937  001132' 005700                    TST     R0                      ;DATA RECEIVED?
2938  001134' 001402                    BEQ     20$                     ;Y,N-20$
2939  001136' 010167  011156'           MOV     R1,CTFCAR               ;STORE NEW FILL CHAR
2940  001142' 122703  000012    20$:    CMPB    #012,R3                 ;LINE FEED TERMINATOR?
2941  001146' 001440                    BEQ     RELGBL                  ;N,Y-RELGBL
2942  001150' 004567  005654'  25$:     JSR     R5,MES                  ;ISSUE THIRD MSG
2943  001154' 001632'                   .WORD   CON3MG
2944  001156' 004767  001240'           JSR     PC,OATOB                ;GET REPLY & CONVERT
2945  001162' 000772                    BR      25$                     ;ERROR?  (N,Y-25$)
2946  001164' 005700                    TST     R0                      ;DATA RECEIVED?
2947  001166' 001402                    BEQ     30$                     ;Y,N-30$
2948  001170' 010167  011154'           MOV     R1,CTFCNT               ;STORE NEW FILL CNT
2949  001174' 122703  000012    30$:    CMPB    #012,R3                 ;LINE FEED TERMINATOR?
2950  001200' 001423                    BEQ     RELGBL                  ;N,Y-RELGBL
2951  001202' 004567  005654'  35$:     JSR     R5,MES                  ;ISSUE LOWER CASE MSG
2952  001206' 001657'                   .WORD   CON4MG
2953  001210' 004567  007124'           JSR     R5,CTRD                 ;ISSUE RD FOR REPLY
2954  001214' 000100                    .WORD   EKYBUF-.
2955  001216' 000025                    .WORD   21.
2956  001220' 004767  001376'           JSR     PC,CKRPLY               ;GO CHECK THE REPLY
2957  001224' 000766                    BR      35$                     ;BR IF INV DATA
2958  001226' 000410                    BR      RELGBL                  ;BR IF Y, <CR>, OR <LF>
2959  001230' 012702  010206'           MOV     #LOWCAS,R2              ;NO, SET UP ADR OF INST'S TO WIPE OUT
2960  001234' 012703  000010           MOV     #8.,R3                  ;SET UP # OF WORDS
2961  001240' 012722  000240    50$:    MOV     #240,(R2)+              ;CLEAR INST'S TO NOP'S
2962  001244' 005303                    DEC     R3
2963  001246' 001374                    BNE     50$
2964
2965
2966                                     ;RELOCATE EXEC INTERFACE ADR'S
2967
2968  001250' 012700  001424'  RELGBL:  MOV     #EGBLTE,R0              ;GET END ADR OF INTERFACE ADR'S TBL
2969  001254' 016701  000466'           MOV     ENDMEM,R1               ;GET END OF MEM ADR
2970  001260' 012702  000027            MOV     #EGBLTE-EGBLTS/2,R2     ;GET # OF ADDRESSES
2971  001264' 010267  000076'           MOV     R2,ESGCNT               ;STORE THIS CNT FOR HSKP EXTENSION
2972  001270' 014041           85$:     MOV     -(R0),-(R1)             ;STORE ADR AT TOP OF MEM
2973  001272' 005302                    DEC     R2                      ;DECR ADDRESS CNT
2974  001274' 001375                    BNE     85$                     ;STORED ALL? (Y,N-85$)
2975  001276' 010167  000062'           MOV     R1,ESGADR               ;STORE ADR OF STORED ADR'S
2976
2977
2978            001                     .IF DF MIMIC
2979                                    ;DISPLAY SLEEP BREAKPOINT ADDRESS
2980
2981                             DSBKPT: MOV     #SLPBKP,R1              ;GET ADR OF BREAKPOINT INST
2982                                     MOV     #SBKPAD,R2              ;GET ADR OF MSG AREA
2983                                     JSR     PC,OBTOA                ;CONVERT ADR TO BIN & PUT IN MSG
2984                                     JSR     R5,MES                  ;ISSUE SLP BKPT ADR MSG
2985                                     .WORD   SBKPMG
2986
2987
```

```
                        000                             .ENDC
                                                        ;HOUSEKEEPING EXIT

001302' 012745  000240          EXHKEX: MOV     #240,-(R5)              ;SET CALLING JSR TO NOP'S
001306' 011545                          MOV     (R5),-(R5)
001310' 000205                          RTS     R5                      ;EXIT IN-LINE

001312' 001314'                 EKBPTR: .WORD   EKYBUF                  ;KEYBOARD DATA BUFFER POINTER
001314' 000025                  EKYBUF: .BLKB   21.                     ;21 BYTE KEYBOARD BUFFER
001341'     015                         .BYTE   015

001342' 000000  000000          LIMIT:  .LIMIT                          ;EXEC'S START & END ADR --
                                                                        ;TAILORED BY THE LNKX11 PROGRAM
                        001             .IF DF MIMIC

                                SBKPMG: .ASCII  <015><012>/SLEEP BKPT = /
                                SBKPAD: .ASCIZ  /XXXXXX/<015><012>
                                        .EVEN
                                        .PAGE
                                        .SBTTL  ROUTINE THAT CREATES DISK FOR MIMIC DEBUGGING


                                        RKER=   177402
                                        RKCS=   177404
                                        RKWC=   177406
                                        RKBA=   177410
                                        RKDA=   177412


                                DISKWR: MOV     #37744,R6               ;INITIALIZE STACK POINTER
                                        MOV     LIMIT+2,R0              ;GET END OF EXEC+MPG ADR
                                        ASR     R0                      ;MAKE IT A WORD COUNT
                                        NEG     R0                      ;MAKE IT NEGATIVE
                                        MOV     R0,(PC)+                ;STORE IT AS EXEC'S WORD COUNT
                                ESIZE:  .WORD   0
                                        MOV     #-256.,@#RKWC           ;SET WORD COUNT TO 1 BLOCK
                                        CLR     @#RKBA                  ;SET BUSS ADR TO 0
                                        CLR     @#RKDA                  ;SET BLOCK # TO 0
                                        MOV     #3,@#RKCS               ;ISSUE WRITE DATA CMND
                                1O$:    TST     @#RKCS                  ;ERROR BIT SET?
                                        BMI     4O$                     ;N,Y-4O$
                                        TSTB    @#RKCS                  ;DONE BIT SET?
                                        BPL     1O$                     ;Y,N-1O$
                                        MOV     ESIZE,@#RKWC            ;SET UP EXEC'S WORD COUNT
                                        CLR     @#RKBA                  ;SET BUSS ADR TO 0
                                        MOV     #4O,@#RKDA              ;SET BLOCK # TO 3O
                                        MOV     #3,@#RKCS               ;ISSUE WRITE DATA CMND
                                2O$:    TST     @#RKCS                  ;ERROR BIT SET?
                                        BMI     5O$                     ;N,Y-5O$
                                        TSTB    @#RKCS                  ;DONE BIT SET?
                                        BPL     2O$                     ;Y,N-2O$
                                        RESET                           ;CLEAR EVERYTHING
                                        HALT                            ;FINISHED CREATING DISK

                                ;ERROR ENCOUNTERED
```

```
3044
3045                            40$:    CLR     R2                      ;SET FLAG TO THIS ENTRY POINT
3046                                    BR      60$
3047                            50$:    MOV     #-1,R2                  ;SET IT TO THIS ENTRY POINT
3048                            60$:    MOV     @#RKCS,R0               ;SAVE CONTROL STATUS
3049                                    MOV     @#RKER,R1               ;SAVE ERROR REG
3050                                    MOV     #1,@#RKCS               ;ISSUE CONTROLLER RESET CMND
3051                                    HALT                            ;ERROR HALT
3052                                    BR      DISKWR                  ;GO RETRY
3053            DOC                     .ENDC
```

```
                                        .SBTTL  MPG/EXEC INTERFACE ADDRESSES TABLE

001346'                         EGBLTS:
001346' 003656'                         .WORD   BOOT
001350' 001144'                         .WORD   BUF
001352' 000000G                         .WORD   CFTCHI
001354' 011436'                         .WORD   CLISTI
001356' 003136'                         .WORD   CLOSE
001360' 007256'                         .WORD   CPRT
001362' 003112'                         .WORD   CREATE
001364' 000000G                         .WORD   CSAVEI
001366' 011056'                         .WORD   CTFLGW
001370' 007124'                         .WORD   CTRD
001372' 010450'                         .WORD   CTRDEX
001374' 007104'                         .WORD   CTRDNE
001376' 006646'                         .WORD   CTUSWR
001400' 006750'                         .WORD   CTWR
001402' 007306'                         .WORD   CUSPRT
001404' 003514'                         .WORD   DELETE
001406' 002412'                         .WORD   GET
001410' 003576'                         .WORD   LIST
001412' 003244'                         .WORD   LOADVR
001414' 003006'                         .WORD   OPENL
001416' 003142'                         .WORD   PUT
001420' 003542'                         .WORD   ZERO
001422' 000536'                         .WORD   REALNM
        001424'                 EGBLTE  =.


    001                                 .IF DF MIMIC

                                        .BLKW   1.                      ;MODIFIABLE FILL AREA USED
                                                                        ;TO FORCE MPG START ADR
                                                                        ;OF 20000 WHEN UNDER MIMIC
                                                                        ;AND MEM MGMNT.

                                UFWSET: .WORD   XXXX
                                UFWCLR: .WORD   XXXX
                                UFWTST: .WORD   XXXX
                                BINASC: .WORD   XXXX
                                BTASLB: .WORD   XXXX
                                CSYSFW: .WORD   USRINT
                                USRINT: .WORD   XXXX

                                MPGBGN  =.

                                ALPT2=  .-HKLOCZ
                                ALFCT2= ALPT2&000077/2
    002                                 .IF NE ALFCT2
                                ALQUN2= 31.-ALFCT2
                                        .BLKW   ALQUN2                  ;ALIGN TO NEXT 32 WORD BNDRY
                                        .WORD   0
    001                                 .ENDC
    000                                 .ENDC
```

```
3111            001424'                 EXCEND  =.
3112
3113            000000'                         .END    EXECST
```

```
ALC    = 177772          CPRTFG  011204R     003   CTRTRM  010726R     003   EGETWD  002714R     003   EUSRSA  003000R     003
ATOI     001144R     003 CPRTRM  011340R     003   CTRVEC  011142R     003   EGTDAT  000256R     005   EWRITL  004534R     003
BCDCV    005730R   G 003 CPRUFW  011434R     003   CTUDED= 020000            EHKBAT  000236R     005   EXCEND= 001424R     005
BCLEAR   005000R   G 003 CPRUSR= 100000            CTUSWR  006646R     003   EHKMEM  000174R     005   EXCID  = ******   G
BINASC   000006R     005 CREATE  003112R     003   CTWABT  007720R     003   EINSFM  000244R     005   EXCIDT  002040R     003
BKRDO    004516R     003 CREGSV  004172R     003   CTWADR  011062R     003   EINTOL  004312R     003   EXDRET  003066R     003
BKREAD   004504R   G 003 CRLF    005546R   G 003   CTWAIT= 002000            EJNFND  003106R     003   EXECST  000000R   G 003
BKWRTO   004652R     003 CRPMAP  005010R   G 003   CTWBR   011152R     003   EJNRET  003536R     003   EXERET  003072R     003
BKWRTI   004676R     003 CRTREG  011040R     003   CTWBSY= 000002            EKBDSK  000536R     005   EXERM   006604R     003
BLKWRT   004644R   G 003 CRUDED= 040000            CTWCNT  011060R     003   EKBPTR  001312R     005   EXERPT  006532R     003
BMOVE    004744R   G 003 CSAVEI= ******   G       CTWCOM  007012R     003   EKYBUF  001314R     005   EXHKEX  001302R     005
BOOT     003656R     003 CSVREG  011024R     003   CTWDRA  011146R     003   EMCOM   004752R     003   EXNRET  003056R     003
BTASLB   000010R     005 CSYSFW  000012R     005   CTWFOT  007652R     003   EMDLTB  012006R     003   EXYBK   004666R     003
BUF      001144R   G 003 CTBRET  006730R     003   CTWINT  007626R     003   EMESEX  005722R     003   E1THKP  000016R     005
BUFEND   002144R     003 CTCKCH  010044R     003   CTWPEB  010754R     003   EMFND   000220R     005   FCLOSE  004704R     003
CFTCHI= ******   G       CTCONS  001056R     005   CTWR    006750R     003   EMINST  004764R     003   FLNOTF  006356R   G 003
CHKSUM   000520R     003 CTCREQ= 001000            CTWRCD  007710R     003   EMONMG  006206R     003   FMTVCT  000132R     005
CHPINT= 004000            CTDOOT  010014R     003   CTWRCE  007656R     003   EMONTB  006214R     003   FSTMOD  000515R   G 003
CHROUT   005512R   G 003 CTDUWR  006740R     003   CTWREX  010006R     003   EMSADR  002774R     003   GET     002412R     003
CKRPLY   001376R     003 CTEBFL= 000024            CTWUFW  011064R     003   ENDMEM  000466R     003   HKLOCZ  000000R     005
CKSMER   006444R     003 CTEBSY= 000040            CTWUSR= 100000            ENMRET  004432R     003   IBKLGT  002200R     003
CK4MM    000702R     005 CTEBUF  011114R     003   CTWVEC  011150R     003   ENXDTA  002776R     003   IDHSU   002146R     003
CLIST    007240R     003 CTECNT  011106R     003   CUSPGR= 002000            EOFMSG  006626R     003   IFLCNT  002154R     003
CLISTI   011436R     003 CTEIAD  011104R     003   CUSPRT  007306R     003   EPARBK  003004R     003   IFLMOD  002152R     003
CLOSE    003136R     003 CTFCAR  011156R     003   DATELF  001060R     005   EPKCNT  005200R     003   IFNAM   002256R     003
CLRBUF   005024R   G 003 CTFCNT  011154R     003   DATUPK  006036R   G 003   EPKCOM  005202R     003   ILSTBK  002202R     003
CLS    = 177764          CTFID   011160R     003   DDBEND  002412R     003   EPKEX   005320R     003   INALC   002224R     003
CMPNAM   005040R   G 003 CTFLGW  011056R     003   DDBSTR  002150R     003   EPKPEX  005314R     003   INBA    002236R     003
CNTRLO= 010000            CTFSCH  011100R     003   DELAY   005124R   G 003   ERDEOF  003005R     003   INBKCT  002250R     003
CON1MG   001522R     003 CTFTCT  011076R     003   DELERR  006274R   G 003   ERDERR  004440R     003   INBLK   002240R     003
CON2MG   001605R     003 CTPRIO= 000020            DELETE  003514R     003   ERDLNK  004400R     003   INBOOT  002206R     003
CON3MG   001632R     003 CTRADR  011070R     003   DELFIL  006310R   G 003   ERET    004436R     003   INCLS   002216R     003
CON4MG   001657R     003 CTRBR   011144R     003   DEVERR  006324R   G 003   ERMFDN  006613R     003   INCM    002232R     003
CPRADR   011430R     003 CTRBSY= 000001            DEVFUL  006340R   G 003   ERRCOM  006462R     003   INCOM   002242R     003
CPRBSY= 000001            CTRCNT  011066R     003   DIRBLK= ******   G       ESGADR  000062R     003   INDEND= 002270R     003
CPRCNT   011432R     003 CTRCOM  007142R     003   DLT    = 177762           ESGCNT  000076R     003   INDEV   002232R     003
CPRCOM   007372R     003 CTRCRQ  010622R     003   DRT    = 177756           ESUCOM  004126R     003   INDIR   002252R     003
CPRCTC   011256R     003 CTRCTU  010534R     003   EBCDMA  005776R     003   ESUFNM  004052R     003   INDLT   002214R     003
CPRDOT   011332R     003 CTRD    007124R     003   ECFORC  006474R     003   ESUFTH  004030R     003   INDRT   002210R     003
CPRERR= 000200            CTRDEC= 000200            ECOEFF  005504R     003   ESUFTI  004034R     003   INDRV   002230R     003
CPREX    011250R     003 CTRDEX  010450R     003   ECOMEX  004772R     003   ESUIF   004104R     003   INDT    002240R     003
CPRFWD   011426R     003 CTRDME  007104R     003   ECOMTY  005516R     003   ESUIOC  004230R     003   INETR   002220R     003
CPRIGN   011210R     003 CTRDON= 000400            EDATER  000570R     005   ESUIPT  004216R     003   INNBK   002254R     003
CPRINT   011162R     003 CTRDRA  011140R     003   EDERMG  001504R     003   ESUOPT  004224R     003   INPRC   002244R     003
CPRLNC   011232R     003 CTRECL  010552R     003   EDMSG   001454R     003   ESUOS   004116R     003   INPWC   002246R     003
CPRMCK   011230R     003 CTRINT  010162R     003   EDTACT  002772R     003   ESUSAV  003762R     003   INSRH   002222R     003
CPRRTC   011314R     003 CTRLC   010566R     003   EFDEVN  006642R     003   ESYDSU  000600R     005   INSRV   002226R     003
CPRSCH   011244R     003 CTRPEB  010746R     003   EFMSZ   000050R     003   ETMP    002722R     003   INVCMD  006416R     003
CPRSTC   011306R     003 CTRPEX  011022R     003   EGBLTE= 001424R     005   ETR    = 177766           INVDMG  001725R     003
CPRT     007256R     003 CTRQBP= 100000            EGBLTS  001346R     005   ETYEX   006030R     003   INWC    002234R     003
CPRTAB   011324R     003 CTRRKY  010456R     003   EGERET  002754R     003   ETYPWK  005534R     003   INZER   002212R     003
CPRTBC   011240R     003 CTRRUB= 000100            EGETBY  002742R     003   EUPKCM  005372R     003   ISUFM   000136R     003
CPRTER   007600R     003 CTRTRB  007222R     003   EGETKG  002762R     003   EUSRCT  003002R     003   ISVBLK  002162R     003
```

```
ISVCNT  002160R     003   MMVER = 000001
ISVDAT  002172R     003   MPGADR  000046R     003
ISVEXT  002170R     003   MPGBGN= 000016R     005
ISVMAP  002156R     003   MPGID   000034R     003
ISVNAM  002164R     003   MPGST   000472R     003
ISVUPT  002204R     003   NFNAME  006364R     003
ISVXX   002174R     003   NXTBLK  004452R   G 003
ITOA    006004R   G 003   OATOB   001240R     003
IWCTR   002150R     003   OBKLGT  002322R     003
I1STBK  002176R     003   ODHSU   002270R     003
LDEVER  003474R     003   OFLCNT  002276R     003
LDRBYC  003252R     003   OFLMOD  002274R     003
LDRDTA  003354R     003   OFNAM   002400R     003
LDRDTC  003356R     003   OLSTBK  002324R     003
LDREXC  003432R     003   OPENL   003006R     003
LDRGET  003346R     003   OSVBLK  002304R     003
LDRISM  003504R     003   OSVCNT  002302R     003
LDRMST  003246R     003   OSVDAT  002314R     003
LDRNEX  003424R     003   OSVEXT  002312R     003
LDROPN  003012R     003   OSVMAP  002300R     003
LDUNUM  000604R   G 005   OSVNAM  002306R     003
LDUREQ  001745R     003   OSVUPT  002326R     003
LIGHTS  000400R     003   OSVXX   002316R     003
LIMIT   001342R     005   OUALC   002346R     003
LIST    003576R     003   OUBKCT  002372R     003
LMPGDE  000132R     003   OUBOOT  002330R     003
LMPGIM  000124R     003   OUCLS   002340R     003
LMPGIN  000060R     003   OUDLT   002336R     003
LOADVR  003244R     003   OUDRT   002332R     003
LOC600  000000R     003   OUETR   002342R     003
LOWCAS  010206R     003   OUPRC   002366R     003
LSTFLG  000514R   G 003   OUPWC   002370R     003
LTSEX   000464R     003   OUSRH   002344R     003
LTSPAT  000402R     003   OUTBA   002360R     003
L2MCNT  000424R     003   OUTBLK  002362R     003
MAPBUF  000544R   G 003   OUTCM   002354R     003
MBUF    000744R   G 003   OUTCOM  002364R     003
MES     005654R   G 003   OUTDEV  002352R     003
MMAP    000754R   G 003   OUTDIR  002374R     003
MMCKEX  001040R     005   OUTDRV  002352R     003
MMCKRT  001014R     005   OUTDT   002362R     003
MMMR0 = 177572            OUTNBK  002376R     003
MMMR3 = 172516            OUTSRV  002350R     003
OUZER   002334R     003   PTWR    011520R     003
OWCTR   002272R     003   PTZERO  011532R     003
O1STBK  002320R     003   PT1     011640R     003
PACK    005164R     003   PT2     011646R     003
PACK1   005174R     003   PT3     011702R     003
PAKNAM  005136R   G 003   PUT     003142R     003
PC    =%000007            READBK  004524R   G 003
PIPFLG  000522R   G 003   REALNM  000536R     003
POSADJ  000120R     003   RELGBL  001250R     005
PRTDRA  011446R     003   RST04   004730R   G 003
PRTFLG  000516R     003   R0    =%000000
PRTIVA  011450R     003   R1    =%000001
PRTPSW  011452R     003   R2    =%000002
PTALLC  011550R     003   R3    =%000003
PTBADR  011510R     003   R4    =%000004
PTBLK   011512R     003   R5    =%000005
PTBOOT  011532R     003   R6    =%000006
PTCLOS  011556R     003   SAV04   004716R   G 003
PTCMND  011514R     003   SETDED= 000040
PTCNT   011732R     003   SETPP   011570R     003
PTCOM   011576R     003   SETPR   011560R     003
PTCREG  011504R     003   SP    =%000006
PTDBLK  011530R     003   SRH   = 177770
PTDELT  011532R     003   STACK = 001200
PTDIR   011532R     003   STKADR  000512R     003
PTDONE  011722R     003   TAB     005564R   G 003
PTDPTR  011524R     003   TODAY   000470R     003
PTDVR   011604R     003   TTYERR  000114R     005
PTENT   011550R     003   TTYTRP  000124R     005
PTERR   012002R     003   TTYTST  000100R     005
PTIMER  011730R     003   TXEXT   000532R   G 003
PTIMEX  012000R     003   TXNAM   000524R   G 003
PTIME1  011734R     003   UFWCLR  000002R     005
PTLBLK  011526R     003   UFWHLD  011456R     003
PTLOOK  011536R     003   UFWSET  000000R     005
PTNOUT  011606R     003   UFWTST  000004R     005
PTOUT   011774R     003   UNIMAP= 000040
PTPARM  011460R     003   UNPACK  005354R     003
PTPEND  011530R     003   UPACK1  005364R     003
PTRBCT  011522R     003   UPKNAM  005326R   G 003
PTRD    011516R     003   USEMM   002006R     003
PTSUAD  012010R     003   USRINT  000014R     005
PTUNIT  011502R     003   WRTBLK  004660R   G 003
WRTLC   004632R   G 003
XBA   = 000004
XBC   = 000016
XBKLGT= 177746
XBT   = 177754
XCM   = 000000
XCO   = 000010
XDHSU = 177714
XDN   = 177776
XDR   = 000020
XDT   = 000006
XFLCNT= 177722
XFLMOD= 177720
XLSTBK= 177750
XNB   = 000022
XRD   = 000012
XSV   = 177774
XSVBLK= 177730
XSVCNT= 177726
XSVDAT= 177740
XSVEXT= 177736
XSVMAP= 177724
XSVNAM= 177732
XSVUPT= 177752
XSVXX = 177742
XWC   = 000002
XWCTR = 177716
XWT   = 000014
XXNAM = 000024
XXXX  = 000000
X1STBK= 177744
ZER   = 177760
ZERO    003542R     003
$BUF    000474R   G 003
$BUF10  000500R   G 003
$BUF2   000476R   G 003
$MAPBF  000502R   G 003
$MBUF   000504R   G 003
$MMAP   000506R   G 003
$TPNM1  005600R   G 003
$TXNAM  000510R   G 003
$TYPNM  005574R   G 003
.     = 001424R     005

. ABS.  000000    000
        000000    001
LOADER  000000    002
EXEC    012016    003
HANDLR  000000    004
EXHSKP  001424    005

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DTECAB/NL:TOC/DOC=DTECAB.P11
```

RUN-TIME: 9 19 1 SECONDS
RUN-TIME RATIO: 54/30=1.7
CORE USED: 6K (11 PAGES)

DOCUMENT PAGES: 72

Spooler runtime 10 Seconds, [illegible] disk reads, 3 disk writes, 72 pages