PDP11

PROGRAM GENERATOR

MD-11-DTMGA-C

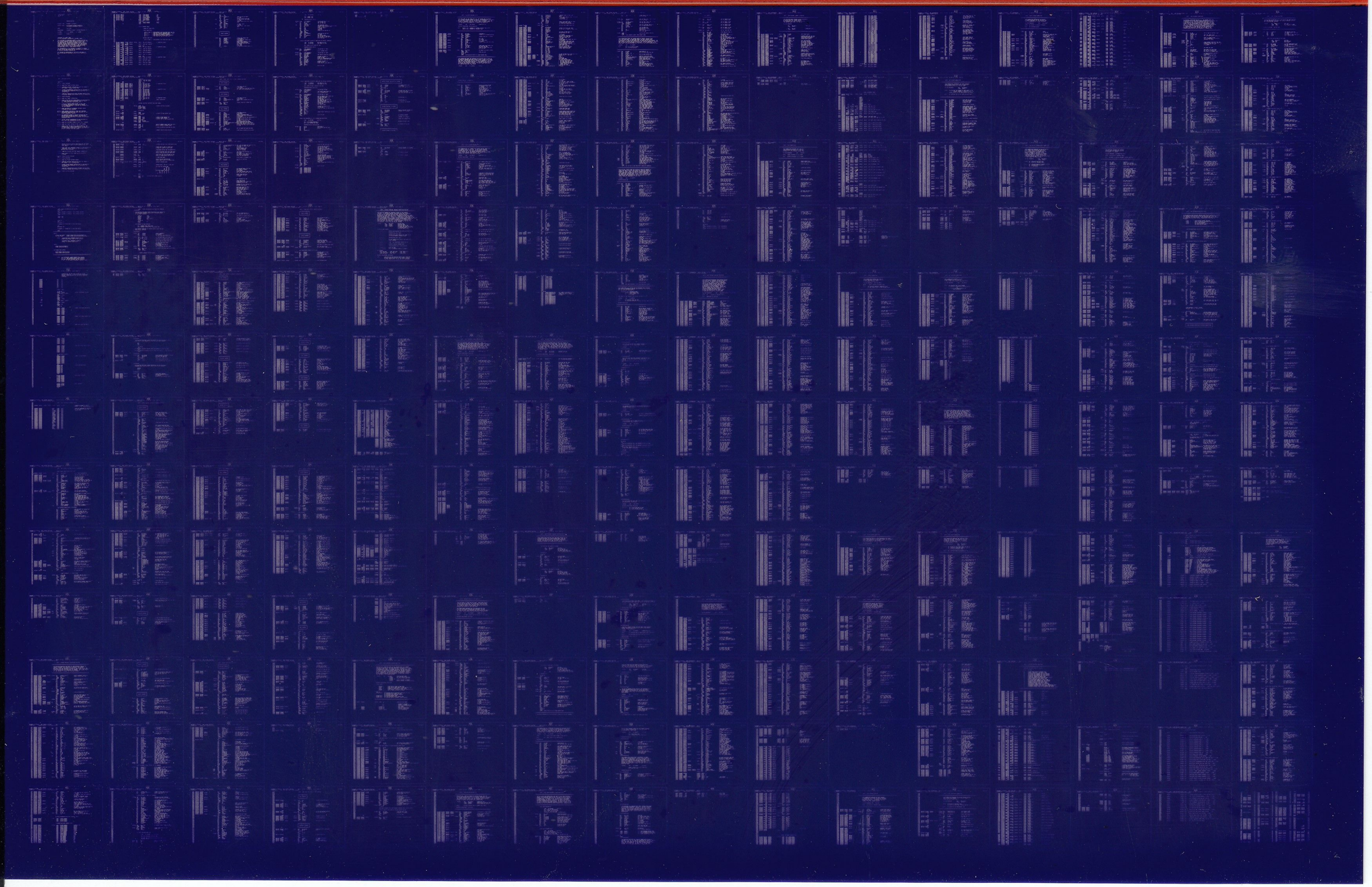
EP-DTMGA-C-DL-A

FICHE 1 OF 2

NOV 1976

digital

MADE IN USA



PDP11

PROGRAM GENERATOR

MD-11-DTMGA-C

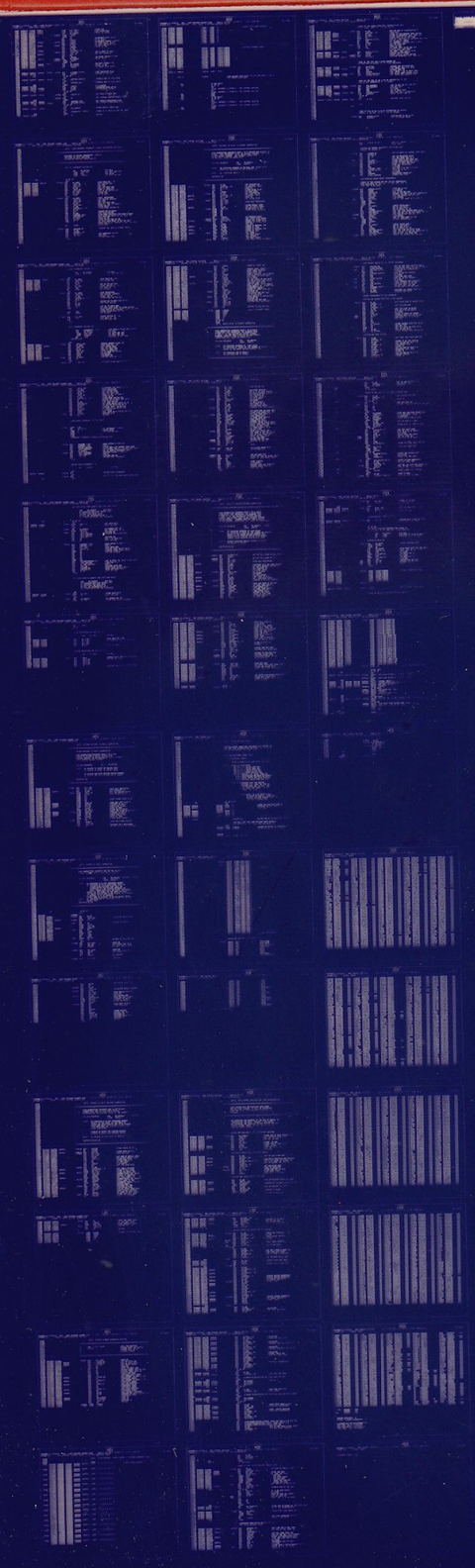
EP-DTMGA-C-DL-A

FICHE 2 OF 2

NOV 1976

digital

MADE IN USA



```
                001
                                .IIF DF NLIST, .NLIST CND
                                .IF NDF MM
                                .REM    %
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTMGA-C

PRODUCT NAME: MAINTENANCE PROGRAM GENERATOR
NON-MEMORY MANAGEMENT VERSION

DATE: JULY 1976

MAINTAINER: DIAGNOSTIC GROUP / SYSTEMS RELIABILITY

AUTHORS: W. R. GREENE / C. E. HARPER

DOCUMENT: DTUMA-C-D


```
                                %
                000             .ENDC
```

```
                        .SBTTL  REVISION HISTORY

                        .IF DF MM
                        .IFF
001                     JUL 76  DTMGA-C RELEASE

                        MAY 76  ADDED THE "PATB" PATTERN FORMAT.

                        MAY 76  CORRECTED ANOTHER COMPILER BUG WITH THE "BREAK V"
                                AND "BREAK V ON V" INSTRUCTIONS.

                        MAY 76  CHANGED SIGNED BRANCH INSTRUCTIONS TO UNSIGNED ON
                                THE BINARY CODE GENERATED FOR ALL RELATIVE "IF"
                                INSTRUCTIONS.

                        MAY 76  CORRECTED COMPILER BUG WHICH OCCURRED WHEN AN OCTAL
                                OR DECIMAL ADDRESS WAS ENTERED ON AN I/O INSTRUCTION.

                        APR 76  DTMGA-B RELEASE

                        FEB 76  ADDED THE "WORD" INSTRUCTION.

                        JAN 76  ADDED A TEST FOR THE CACHE MEMORY HIT/MISS REGISTER.
                                IF FOUND, THE DEVICE ROUTINES' TIME OUT VALUE
                                WILL BE EFFECTIVELY MULTIPLIED BY 2.

                        JAN 76  IF AN I/O TIMEOUT ERROR OCCURS, IT WILL NOW
                                BE TREATED THE SAME AS OTHER DEVICE ERRORS. IT
                                IS NO LONGER REQUIRED TO RESTART THE PROGRAM WITH
                                THE RUN COMMAND.

                        DEC 75  ADDED THE ABILITY TO PROCESS VARIABLE LENGTH
                                FILENAMES ON USER PROGRAMS.

                        DEC 75  ADDED SEVERAL SECTIONS OF CODE THAT PERFORMS
                        NOV 75  EQUIVALENT FUNCTIONS THAT WERE REQUIRED FOR THE
                                MEMORY MANAGEMENT VERSION.

                        DEC 75  BUS ADDRESSES PASSED TO DEVICE ROUTINES FOR NPR
                                DEVICES WILL NOW BE TWO WORDS. THIS PROVIDES
                                DEVICE ROUTINE COMPATIBILITY BETWEEN THE DIFFERENT
                                VERSIONS OF MPG.

                        NOV 75  INITIAL VECTOR HOUSEKEEPING IS NOW PERFORMED BY
                                MPG'S ONE TIME HOUSEKEEPING ROUTINE.

                        NOV 75  ADDED THE "PATTERN" FORMAT TO THE 'FILL' COMMAND.

                        NOV 75  ADDED THE "FAST" OPTION TO THE /LIST COMMAND.

                        NOV 75  ADDED A TEST FOR THE LSI-11 AND ALTERED ALL PSW
                                ACCESSES FOR 11 FAMILY COMPATIBILITY. IF AN LSI -11,
                                ALL BUS REQUEST PRIORITIES IN DEVICE ROUTINES WILL
                                BE SET TO 4 WHEN THEY ARE LOADED.
```

```
                                ;
                                ;       NOV 75  REMOVED THE VALID DEVICE TABLE FROM MPG AND ADDED
                                ;               CODE THAT WILL LOAD IT FROM A SEPARATE FILE INTO
                                ;               MPG'S AREA.
                                ;
                                ;       OCT 75  ADDED CODE THAT TRANSFERS CONTROL TO A DEVICE ROUTINE
                                ;               IMMEDIATELY AFTER LOADING IT FOR INITIAL HOUSE-
                                ;               KEEPING AND TAILORING.
                                ;
                                ;       OCT 75  CORRECTED THE BUG WITH THE "BREAK" INSTRUCTION.
                                ;
                                ;       OCT 75  ADDED DISPLAY OF THE LOADED DEVICE ROUTINE'S
                                ;               FILENAME.
                                ;
                                ;       AUG 75  DTMGA-A INITIAL RELEASE
                                .IFT
                                ;       JUL 76  DTMMA-B RELEASE
                                ;
                                ;       MAY 76  ADDED THE "PATB" PATTERN FORMAT.
                                ;
                                ;       MAY 76  CORRECTED ANOTHER COMPILER BUG WITH THE "BREAK V"
                                ;               AND "BREAK V ON V" INSTRUCTIONS.
                                ;
                                ;       MAY 76  CHANGED SIGNED BRANCH INSTRUCTIONS TO UNSIGNED ON
                                ;               THE BINARY CODE GENERATED FOR ALL RELATIVE "IF"
                                ;               INSTRUCTIONS.
                                ;
                                ;       MAY 76  CORRECTED COMPILER BUG WHICH OCCURRED WHEN AN OCTAL
                                ;               OR DECIMAL ADDRESS WAS ENTERED ON AN I/O INSTRUCTION.
                                ;
                                ;       APR 76  DTMMA-A INITIAL RELEASE
                                .ENDC
000
```

```
                                .SBTTL  BUILD INFORMATION AND EQUATES
     001                        .IF DF MM
                                .IFF
                                .TITLE  MAINDEC-11-DTMGA-C   MPG CONTROL ROUTINE
                                .IFT
                                .TITLE  MAINDEC-11-DTMMA-B   MPG CONTROL ROUTINE
     000                        .ENDC

  000000'                       .CSECT  MPG
                                .DSABL  GBL


     001                        .IF DF MM
                                .IFF
                                ;REVISION "C"

                                ;FILENAME OF "TMGAC0.MPG" ON MPG/XXDP MEDIA

                                .IFT
                                ;REVISION "B"

                                ;FILENAME OF "TMMAB0.MPG" ON MPG/XXDP MEDIA

     000                        .ENDC
                        ;*******************************************************************
                        ;
                        ; MACY11:
                        ;
                        ;    WITHOUT MEM MGMNT:  DTMGA?,DTMGA?/CRF:SYM/DOC=DTMGA?.P11
                        ;    WITH MEM MGMNT:     DTMMA?,DTMMA?/CRF:SYM/EQ:MM/DOC=DTMGA?.P11
                        ;
                        ;       NOTES:  "/EQ:NLIST" WILL SUPPRESS DISPLAY OF ALL
                        ;               UNUSED CONDITIONAL CODE IN THE LISTING.
                        ;
                        ;               "/EQ:MIMIC" WILL GENERATE A VERSION
                        ;               CAPABLE OF RUNNING UNDER MIMIC.
                        ;
                        ;
                        ; LNKX11:
                        ;
                        ;    DTMGA?.MPG/B:0←DTMGA?/E
                        ;    DTMMA?.MPG/B:0←DTMMA?/E
                        ;
                        ; FOR PAPER TAPE OUTPUT:
                        ;
                        ;    PUNCH DTMGA?.MPG/FILE:ELEV
                        ;    PUNCH DTMMA?.MPG/FILE:ELEV
                        ;
                        ;*******************************************************************


                            ;NOTE:  ANY STATEMENTS, WHOSE COMMENT FIELD BEGINS
                            ;       WITH ;*;, ARE STORAGE AREAS WHOSE CONTENTS
                            ;       WILL BE DIFFERENT BETWEEN THE TWO VERSIONS
                            ;       OF MPG. ON THE MEMORY MANAGEMENT VERSION, THEIR
```

```
                                ;       CONTENTS WILL BE THE HIGH 16 BITS OF 22 BIT
                                ;       ABSOLUTE ADDRESSES. ON THE NON-MEMORY MANAGEMENT
                                ;       VERSION, THEIR CONTENTS WILL BE 16 BIT ABSOLUTE
                                ;       ADDRESSES.


000000                          R0=     %0
000001                          R1=     %1
000002                          R2=     %2
000003                          R3=     %3
000004                          R4=     %4
000005                          R5=     %5
000006                          R6=     %6
000007                          R7=     %7
000006                          SP=     %6
000007                          PC=     %7

001200                          STACK=  1200

   001                          .IF DF MM
                                MMMR0=  177572                  ;MEMORY MANAGEMENT EQUATES
   002                          .IF DF MIMIC & UMTEST
                                .IFF
                                MMMR1=  177574
                                .IFT
                                MMMR1=  574
   001                          .ENDC
                                MMMR2=  177576
   000                          .ENDC
   001                          .IF DF MIMIC & UMTEST
                                .IFF
172516                          MMMR3=  172516
                                .IFT
                                MMMR3=  576
   000                          .ENDC
   001                          .IF DF MM

                                KPAR0=  172340                  ;KERNEL INST SPACE ADR REG'S
                                KPAR1=  172342
                                KPAR2=  172344
                                KPAR3=  172346
                                KPAR4=  172350
                                KPAR5=  172352
                                KPAR6=  172354
                                KPAR7=  172356

                                KPDR0=  172300                  ;KERNEL INST SPACE DESCR. REG'S
                                KPDR1=  172302
                                KPDR2=  172304
                                KPDR3=  172306
                                KPDR4=  172310
                                KPDR5=  172312
                                KPDR6=  172314
                                KPDR7=  172316

                                UPAR0=  177640                  ;USER INST SPACE ADR REG'S
```

```
                                UPAR1=  177642
                                UPAR2=  177644
                                UPAR3=  177646
                                UPAR4=  177650
                                UPAR5=  177652
                                UPAR6=  177654
                                UPAR7=  177656

                                UPDR0=  177600          ;USER INST SPACE DESCR. REG'S
                                UPDR1=  177602
                                UPDR2=  177604
                                UPDR3=  177606
                                UPDR4=  177610
                                UPDR5=  177612
                                UPDR6=  177614
                                UPDR7=  177616

        002                     .IF DF MIMIC & UMTEST
                                .IFF
                                UBMAP0= 170200          ;UNIBUS MAP REGISTERS
                                UBMAP1= 170204
                                UBMAP2= 170210
                                UBMAP3= 170214
                                UBMAP4= 170220
                                UBMAP5= 170224
                                UBMAP6= 170230
                                UBMAP7= 170234
                                UBMAP8= 170240
                                .IFT
                                UBMAP0= 600             ;UNIBUS MAP REGISTERS
                                UBMAP1= 604
                                UBMAP2= 610
                                UBMAP3= 614
                                UBMAP4= 620
                                UBMAP5= 624
                                UBMAP6= 630
                                UBMAP7= 634
                                UBMAP8= 640
        001                     .ENDC

                                PDCONS= 000000          ;PAGE SELECT EQUATES
                                P1CONS= 020000
                                P2CONS= 040000
                                P3CONS= 060000
                                P4CONS= 100000
                                P5CONS= 120000
                                P6CONS= 140000
                                P7CONS= 160000

                                IOPAGE= 177600          ;MISC. EQUATES
                                PDRCON= 077406
                                USRPSW= 140000
                                KERPSW= 0
        000                     .ENDC
```

```
                                        .SBTTL  EXEC INTERFACE ADDRESS TABLE

000000' 053124  040504  037477  CLOCZ:  .ASCII  'TVDA??'                ;CURRENT FILENAME IN ASCII
                                                                         ;PRESET TO THE 'VALID DEVICES' FILE

000006' 000000                  BOOT:   .WORD   XXXX                     ;ABSOLUTE ADDRESSES OF ROUTINES
000010' 000000                  BUF:    .WORD   XXXX                     ;IN THE EXECUTIVE WHICH ARE
000012' 000000                  CFTCHI: .WORD   XXXX                     ;USED BY MPG.
000014' 000000                  CLISTI: .WORD   XXXX
000016' 000000                  CLOSE:  .WORD   XXXX
000020' 000000                  CPRT:   .WORD   XXXX
000022' 000000                  CREATE: .WORD   XXXX
000024' 000000                  CSAVEI: .WORD   XXXX
000026' 000000                  CTFLGW: .WORD   XXXX
000030' 000000                  CTRD:   .WORD   XXXX
000032' 000000                  CTRDEX: .WORD   XXXX
000034' 000000                  CTRDNE: .WORD   XXXX
000036' 000000                  CTUSWR: .WORD   XXXX
000040' 000000                  CTWR:   .WORD   XXXX
000042' 000000                  CUSPRT: .WORD   XXXX
000044' 000000                  DELETE: .WORD   XXXX
000046' 000000                  GET:    .WORD   XXXX
000050' 000000                  LIST:   .WORD   XXXX
000052' 000000                  LOADVR: .WORD   XXXX
000054' 000000                  OPENL:  .WORD   XXXX
000056' 000000                  PUT:    .WORD   XXXX
000060' 000000                  ZERO:   .WORD   XXXX
000062' 000000                  REALNM: .WORD   XXXX
```

```
                                        .SBTTL  MPG CONTROL ROUTINE INITIAL HOUSEKEEPING


                                        ;ENTRY POINT FROM STAND ALONE EXEC

000064' 004567  043476          START:  JSR     R5,C1THKP               ;DO 1 TIME HOUSEKEEPING

000070' 010700                  RST1:   MOV     PC,R0                   ;SET UP CNTRL ROUT TBL ADR
000072' 062700  034124                  ADD     #CRTBL-.,R0
000076' 012702  000020                  MOV     #MAXPRG,R2              ;SET UP # OF PROG'S
000102' 005020                  10$:    CLR     (R0)+                   ;CLEAR TBL ENTRY
000104' 005302                          DEC     R2                      ;DECR PROG COUNT
000106' 001375                          BNE     10$                     ;DONE ALL ENTRIES? (Y,N-10$)
000110' 016767  001340  001330          MOV     CPAREA,CFMST            ;INITIALIZE START OF FREE MEM ADR
000116' 016767  001330  001324          MOV     CRMEND,CFMEND           ;INIT END OF FREE MEMORY ADR
                    001                 .IF DF MM
                                        MOV     CPAREA,R0               ;GET START OF USR PROG AREA
                                        ADD     #577,R0                 ;ALLOW FOR 12K WORDS
                                        CMP     R0,CRMEND               ;ARE THERE MORE THAN 12K WDS?
                                        BHIS    RST2                    ;Y,N-RST2
                                        MOV     R0,CFMEND               ;STORE END OF 12K AREA ADR
                                        CLR     MAPMAP                  ;RESET UNIBUS MAP USAGE
                                        CLR     MAPMAP+2                ;BIT MAP
                    000                 .ENDC
000124' 012706  001200          RST2:   MOV     #STACK,R6               ;SET UP STACK POINTER
000130' 042767  177400  034056          BIC     #177400,CSYSFW          ;INITIALIZE SYSTEM FLGWD
                    001                 .IF DF MM
                                        MOV     #KPAR0,R0               ;SET UP ADR'S OF KERNEL MODE'S
                                        MOV     #KPDR0,R1               ;MM REGS
                                        MOV     #8.,R2                  ;SET UP REGISTER COUNT
                                        CLR     R3                      ;SET MEM BASE ADR TO 0
                                5$:     MOV     R3,(R0)+                ;STORE PAGE'S MEM BASE ADR
                                        MOV     #PDRCON,(R1)+           ;STORE DESCRIPTOR REG VALUE
                                        ADD     #200,R3                 ;POINT BASE ADR TO NXT MEM PAGE
                                        DEC     R2                      ;DONE ALL REGISTERS?
                                        BNE     5$                      ;Y,N-5$
                                        MOV     #IOPAGE,-(R0)           ;SET PAR 7 TO THE I/O PAGE
                                        BIT     #CPU40,CSYSFW           ;IS THIS AN 11/40?
                                        BNE     10$                     ;N,Y-10$
                                        MOV     MR3VAL,@#MMMR3          ;INIT MMMR3 IF UBMAP & 22 BITS
                                10$:    MOV     #1001,@#MMMR0           ;ENABLE MEM MANAGEMENT

                                        ; NOW RUNNING UNDER MEMORY MANAGEMENT


                                        BIT     #UNIMAP,CSYSFW          ;ARE WE USING THE UNIBUS MAP?
                                        BEQ     15$                     ;Y,N-15$
                                        MOV     #UBMAP0,R0              ;POINT AT FIRST MAP REG
                                        CLR     (R0)+                   ;INITIALIZE REGS 0 & 1 TO
                                        CLR     (R0)+                   ;FIRST 8K WDS OF MEM USED
                                        MOV     #20000,(R0)+            ;BY EXEC & MPG
                                        CLR     (R0)+
                                        ADD     #16.,R0                 ;BYPASS REGS 2 THRU 5
                                        MOV     PC,R1                   ;GET ADR OF SHARED CODE PAGE
                                        ADD     #PG6BA-.,R1
                                        MOV     R1,(R0)+                ;STORE IT IN MAP REG 6
                                        CLR     (R0)
```

```
                    000                         .ENDC
000136' 010700                      15$:    MOV     PC,R0                   ;SET UP ADR OF CNTRL ROUT TBL
000140' 062700  034056                      ADD     #CRTBL-.,R0
                    001                     .IF DF MM
                                            .IFF
000144' 012701  000020                      MOV     #MAXPRG,R1              ;HOUSEKEEP PROG FLGWDS IN CASE
000150' 005720                      20$:    TST     (R0)+                   ;WE'RE DOING A RST2 RESTART
000152' 001403                              BEQ     30$
000154' 042750  100070                      BIC     #ACTIVE+WT4IOT+CTPRIO+SETDED,@-(R0)
000160' 005720                              TST     (R0)+
000162' 005301                      30$:    DEC     R1
000164' 001371                              BNE     20$
                                            .IFT
                                    20$:    CMP     #177777,(R0)            ;END OF THE TBL?
                                            BEQ     40$                     ;N,Y-40$
                                            TST     (R0)                    ;ENTRY EMPTY?
                                            BEQ     30$                     ;N,Y-30$
                                            MOV     (R0),@#KPAR6            ;SET PROG'S BASE ADR IN KERNEL'S PAR 6
                                            BIC     #ACTIVE+WT4IOT+CTPRIO+SETDED,@#P6CONS
                                    30$:    TST     (R0)+                   ;POINT TO NXT TBL ENTRY
                                            BR      20$                     ;GO CK NXT ENTRY
                                    40$:    MOV     PC,R0                   ;SET UP ADR OF INTERRUPT
                                            ADD     #VECTBL+4-.,R0          ;INTERCEPT VECTOR TABLE
                                            MOV     #128.,R1                ;# OF ENTRIES
                                    50$:    CLR     (R0)+                   ;CLEAR THIS ENTRY
                                            CLR     (R0)+
                                            ADD     #4,R0                   ;POINT TO NEXT ENTRY
                                            DEC     R1                      ;DONE ALL ENTRIES?
                                            BNE     50$                     ;Y,N-50$
                    000                     .ENDC
000166' 016700  177634                      MOV     CTFLGW,R0               ;GET ADR OF CONSOLE TERM FLGWD
000172' 005020                              CLR     (R0)+                   ;CLEAR THE FLGWD AND WRITE
000174' 005010                              CLR     (R0)                    ;COUNT
000176' 005067  001242                      CLR     CRFLGW                  ;HSKP CNTRL ROUT FLGWD
000202' 012767  000003  000226              MOV     #3,CMGCNT               ;INITIALIZE TO NON-INT MODE
000210' 116767  007120  007115              MOVB    CASMSG+3,CASMSG+2
000216' 004567  010546                      JSR     R5,CSDNRC               ;SHIFT ANY PROGS DOWN & RECOMPILE
                    001                     .IF DF MM
                                            MOV     #-1,CSCFWD              ;INIT TO NO PROG #
                    000                     .ENDC
000222' 000167  004026                      JMP     CMAP                    ;GO DISP MEM MAP


                                            ;USER INTERRUPT (↑C) ENTRY POINT

000226' 012600                      USRINT: MOV     (SP)+,R0                ;GET STACK ADR AT TIME OF INT
000230' 004567  040666                      JSR     R5,BINASC               ;CONVERT IT TO ASCII &
000234' 007614                              .WORD   CINTSP-.                ;PUT IN MSG
000236' 016600  000014                      MOV     12.(SP),R0              ;GET PC VALUE AT INTERRUPT
000242' 004567  040654                      JSR     R5,BINASC               ;CONVERT IT TO ASCII &
000246' 007566                              .WORD   CINTPC-.                ;PUT IN CONSOLE MSG
                    001                     .IF DF MM
                                            MOV     #KPAR4,R0               ;SET UP ADR OF KERNEL WORK PAGES
                                            MOV     (R0)+,-(SP)             ;SAVE WORK PAGES' BASE ADRS
                                            MOV     (R0)+,-(SP)
                                            MOV     (R0),-(SP)
```

```
                                                        MOV     #"KN,CINTMD             ;STORE "KNRL" AS
                                                        MOV     #"RL,CINTMD+2           ;THE MODE
                                                        BIT     #140000,20.(SP)         ;WERE WE IN KERNEL MODE?
                                                        BEQ     10$                     ;N,Y-10$
                                                        MOV     #"US,CINTMD             ;STORE "USER" AS
                                                        MOV     #"ER,CINTMD+2           ;THE MODE
                                                        .ENDC
000250' 005767  034412          10$:    TST     CPGCPT                  ;USER PROG IN CONTROL?
000254' 001412                  BEQ     20$                     ;Y,N-20$
000256' 016704  034210          MOV     CPGCPE,R4               ;GET CURR CNTRL ROUT TBL ENTRY ADR
000262' 004767  013322          JSR     PC,CPNASC               ;CONVERT IT TO ASCII PROG #
000266' 010167  007574          MOV     R1,CINTPN               ;MOVE PROG # TO MSG
000272' 112767  000040  007570  MOVB    #40,CINTPN+2            ;SET THIRD BYTE TO A SPACE
000300' 000406                  BR      30$                     ;GO TO ISSUE MSG
000302' 012767  050115  007556  20$:    MOV     #"MP,CINTPN             ;SET UP "MPG" I.D.
000310' 112767  000107  007552  MOVB    #'G,CINTPN+2
000316' 042767  010000  033670  30$:    BIC     #CNTRLO,CSYSFW          ;RESET SUPPRESS TYPING FLAG
000324' 004567  005674          JSR     R5,CTYPE                ;ISSUE PC & SP VALUES MSG
000330' 000400                  .WORD   400
000332' 007464                  .WORD   CINTMG-.
000334' 012767  000004  000074  MOV     #4,CMGCNT               ;SET UP CNT FOR "I*"
000342' 112767  000111  006763  MOVB    #'I,CASMSG+2            ;PROMPTING MSG
                                                        ;DROP INTO COMMAND DECODER
```

```
                                .SBTTL  COMMAND DECODE & DISPATCHER ROUTINE

                        ;*****************************************************************
                        ;
                        ;  THIS ROUTINE REQUESTS THE ENTRY OF USER COMMANDS THROUGH THE
                        ;  CONSOLE KEYBOARD. UPON RECEIVING A COMMAND, THE COMMAND
                        ;  NAME WILL FIRST BE VALIDATED FOLLOWED BY THE EXTRACTION OF THE
                        ;  PROCESSING ROUTINE ADDRESSES FOR THE COMMAND.  NEXT, THE COMMAND
                        ;  TYPE IS TESTED AND IF IT REQUIRES AN OPERAND, THE COMMAND DATA
                        ;  WILL BE CHECKED TO INSURE THAT ONE WAS ENTERED.
                        ;
                        ;*****************************************************************

000350' 042767 010000 033636  CFCMND: BIC     #CNTRLO,CSYSFW          ;RESET SUPPRESS TYPING FLG
000356' 004567 005642                 JSR     R5,CTYPE                ;ISSUE 'ENTER CMND' MSG
000362' 000400                        .WORD   400
000364' 006730                        .WORD   CECMSG-.
000366' 010700                CNCMND: MOV     PC,R0                   ;INITIALIZE RETURN ADR FOR
000370' 005740                        TST     -(R0)                   ;ERROR ROUTINE
000372' 010067 001204                 MOV     R0,CERRET
000376' 042767 010000 033610          BIC     #CNTRLO,CSYSFW          ;RESET SUPPRESS TYPING FLG
000404' 005767 033604                 TST     CSYSFW                  ;DOING A CONS REQ?
000410' 100406                        BMI     1$                      ;N,Y-1$
000412' 022706 001200                 CMP     #STACK,SP               ;CHECK FOR STACK ERROR
000416' 001403                        BEQ     1$                      ;Y,N-1$
000420' 000000                        HALT
000422' 012706 001200                 MOV     #STACK,R6               ;RE-INITIALIZE STACK PNTR
000426' 004567 005572         1$:     JSR     R5,CTYPE                ;ISSUE "*" W/O TRAILING CR/LF
000432' 000420                        .WORD   420
000434' 006675                        .WORD   CASMSG-.
000436' 000003                CMGCNT: .WORD   3
000440' 004577 177364                 JSR     R5,@CTRD                ;ISSUE KEYBOARD READ
000444' 001020                        .WORD   CCMDRD-.
000446' 000107                        .WORD   71.
000450' 004567 007542                 JSR     R5,CSCNSP               ;SCAN FOR FIRST WORD OF CMND
000454' 010700                        MOV     PC,R0                   ;SET UP ADR OF CONTROL
000456' 062700 000766                 ADD     #CRFLGW-.,R0            ;ROUTINE FLAGWORD
000462' 032702 000010                 BIT     #CSCFND,R2              ;ANY DATA ENTERED ON THE CMND?
000466' 001020                        BNE     CKCMND                  ;N,Y-CKCMND
000470' 005767 033520         CGORUN: TST     CSYSFW                  ;CONSOLE REQ BEING PROCESSED?
000474' 100402                        BMI     2$                      ;N,Y-2$
000476' 000167 033732                 JMP     CPGDSP                  ;GO TO PROGRAM DISPATCHER
000502' 042767 100000 033504  2$:     BIC     #CTRQBP,CSYSFW          ;RESET CT REQ BEING PROC FLAG
000510' 012767 000003 177720          MOV     #3,CMGCNT               ;RESTORE  PROMPTING MSG TO "*"
000516' 116767 006612 006607          MOVB    CASMSG+3,CASMSG+2
                    001               .IF DF MM
                                      MOV     #KPAR6,R0               ;SET UP ADR OF KERNEL WORK PAGES
                                      MOV     (SP)+,(R0)              ;RELOAD ORIGINAL VALUES
                                      MOV     (SP)+,-(R0)
                                      MOV     (SP)+,-(R0)
                                      .ENDC
000524' 000177 177302                 JMP     @CTRDEX                 ;GO RESTORE & EXIT FROM INT
000530' 052710 100000         CKCMND: BIS     #CDDFP,(R0)             ;SET FIRST PASS FLAG
000534' 010702                10$:    MOV     PC,R2                   ;SET UP START ADR OF VALID
000536' 062702 000424                 ADD     #CMDTBL-.,R2            ;CMND TBL
```

```
000542' 012705 000007          MOV     #7,R5           ;SET COMPARE COUNT TO 7
000546' 016703 001024  20$:    MOV     CDSTAD,R3       ;GET DATA WORD START ADR
000552' 010501         30$:    MOV     R5,R1           ;MOVE COMPARE CNT TO WORK CNT REG
000554' 005710                 TST     (R0)            ;FIRST PASS FLAG SET?
000556' 100404                 BMI     40$             ;N,Y-40$
000560' 012701 000002          MOV     #2,R1           ;SET WORK CNT TO 2
000564' 010504                 MOV     R5,R4           ;SAVE COMPARE CNT -2
000566' 160104                 SUB     R1,R4
000570' 122712 000377  40$:    CMPB    #377,(R2)       ;IS ENTRY'S CODE BYTE ALL 1'S
000574' 001013                 BNE     60$             ;Y,N-60$
000576' 122705 000002          CMPB    #2,R5           ;THIS LAST GROUP OF CMND'S?
000602' 001403                 BEQ     50$             ;N,Y-50$
000604' 005305                 DEC     R5              ;DECR COMPARE CNT BY 1
000606' 105722                 TSTB    (R2)+           ;ADD 1 TO COMPARE TBL ADR
000610' 000760                 BR      30$             ;GO DO NEXT CMND GROUP
000612' 005710         50$:    TST     (R0)            ;THIS THE FIRST PASS?
000614' 100045                 BPL     100$            ;Y,N-100$
000616' 042710 100000          BIC     #CDDFP,(R0)     ;RESET FIRST PASS FLAG
000622' 000744                 BR      10$             ;DO SECOND PASS WITH 2 CHAR COMPARES
000624' 112246         60$:    MOVB    (R2)+,-(SP)     ;SAVE FIRST CODE BYTE
000626' 000316                 SWAB    (SP)            ;SAVE SECOND CODE BYTE
000630' 112216                 MOVB    (R2)+,(SP)
000632' 005301         70$:    DEC     R1              ;DECR WORK COMPARE CNT BY 1
000634' 122223                 CMPB    (R2)+,(R3)+     ;CMND TBL CHAR = DATA WORD CHAR?
000636' 001406                 BEQ     90$             ;N,Y-90$
000640' 005710                 TST     (R0)            ;FIRST PASS FLAG SET?
000642' 100401                 BMI     80$             ;N,Y-80$
000644' 060402                 ADD     R4,R2           ;ADD CMP CNT -2 TO TBL ADR
000646' 060102         80$:    ADD     R1,R2           ;ADD REMAINING CNT TO TBL ADR
000650' 005726                 TST     (SP)+           ;TAKE CODE BYTES OFF THE STACK
000652' 000735                 BR      20$             ;GO CHECK NEXT TBL ENTRY
000654' 005701         90$:    TST     R1              ;WORK COMPARE CNT = 0?
000656' 001365                 BNE     70$             ;Y,N-70$
000660' 012604                 MOV     (SP)+,R4        ;GET THIS CMND'S SAVED CODE BYTES
000662' 032704 010000          BIT     #10000,R4       ;CMND POSSIBLY HAVE IMBEDDED PROG #?
000666' 001404                 BEQ     95$             ;Y,N-95$
000670' 004567 012576          JSR     R5,CISDNM       ;GO CHECK IF # IS INCLUDED
000674' 005701                 TST     R1              ;WAS THERE A #?
000676' 001020                 BNE     105$            ;N,Y-105$
000700' 122713 000040  95$:    CMPB    #40,(R3)        ;NEXT CHAR AFTER CMND I.D. A SPACE,
000704' 001415                 BEQ     105$            ;COMMA, CR, OR LF?
000706' 122713 000054          CMPB    #054,(R3)       ;N,Y-105$
000712' 001412                 BEQ     105$
000714' 122713 000015          CMPB    #015,(R3)
000720' 001407                 BEQ     105$
000722' 122713 000012          CMPB    #012,(R3)
000726' 001404                 BEQ     105$
000730' 004567 005270  100$:   JSR     R5,CTYPE        ;GO REPORT INV CMND ERROR
000734' 000112                 .WORD   112             ;RETURN AT LABEL "CNCMND"
000736' 005702                 .WORD   CERM04-.
000740' 005767 033250  105$:   TST     CSYSFW          ;CONSOLE REQ BEING PROCESSED?
000744' 100007                 BPL     CNCTU           ;Y,N-CNCTU
000746' 032704 040000          BIT     #40000,R4       ;THIS A RESTRICTED CMND?
000752' 001404                 BEQ     CNCTU           ;Y,N-CNCTU
000754' 004567 005244  CRSCMD: JSR     R5,CTYPE        ;ISSUE 'RESTRICTED CMND' MSG
000760' 000002                 .WORD   002
```

```
000762' 005643                          .WORD   CERM03-.
000764' 032704 020000           CNCTU:  BIT     #20000,R4       ;THIS CMND REQUIRE OPERANDS?
000770' 001007                          BNE     107$            ;Y,N-107$
000772' 005767 007216                   TST     CSCFWD          ;IS THERE AT LEAST ONE OPERAND?
000776' 100004                          BPL     107$            ;N,Y-107$
001000' 004567 005220                   JSR     R5,CTYPE        ;REPORT "NO OPERAND" ERROR
001004' 000042                          .WORD   042             ;RETURNS AT LABEL "CNCMND"
001006' 005663                          .WORD   CERM07-.
001010' 110405                  107$:   MOVB    R4,R5           ;COMPUTE DISPLACEMENT INTO
001012' 060705                          ADD     PC,R5           ;CMND ROUT ADR TBL
001014' 062705 000060                   ADD     #CRATBL-.,R5
001020' 011505                          MOV     (R5),R5         ;GET REL ADR OF CMND ROUT
001022' 060705                          ADD     PC,R5           ;MAKE IT ABSOLUTE
001024' 010567 000556           CBASE2: MOV     R5,CR2ADR       ;STORE CMND ROUT ADR AT STD LOC
001030' 005704                          TST     R4              ;THIS CMND USE A COM PROC ROUTINE?
001032' 100001                          BPL     110$            ;N,Y-110$
001034' 000115                          JMP     (R5)            ;GO DIRECTLY TO THIS CMND'S ROUTINE
001036' 000304                  110$:   SWAB    R4              ;COMPUTE DISPLACEMENT INTO COM
001040' 042704 177760                   BIC     #177760,R4      ;PROC ROUT ADR TBL
001044' 060704                          ADD     PC,R4
001046' 062704 000016                   ADD     #CPATBL-.,R4
001052' 011404                          MOV     (R4),R4         ;GET REL ADR OF COM PROC ROUT
001054' 060704                          ADD     PC,R4           ;MAKE IT ABSOLUTE
001056' 010467 000522           CBASE1: MOV     R4,CR1ADR       ;STORE THIS ADR AT STD LOC
001062' 000114                          JMP     (R4)            ;GO TO CMND'S COM PROC ROUTINE


                                ;COMMON PROCESSING ROUTINES ADDRESS TABLE

001064' 000632                  CPATBL: .WORD   CCFMT1-CBASE1
001066' 000726                          .WORD   CCFMT2-CBASE1
001070' 000742                          .WORD   CCFMT3-CBASE1
001072' 000620                          .WORD   CCFMT4-CBASE1

                                ;COMMAND ROUTINES ADDRESS TABLE

001074' 002214                  CRATBL: .WORD   CDISPL-CBASE2           ;DISPLAY CMND
001076' 004254                          .WORD   CASSIG-CBASE2           ;ASSIGN
001100' 001370                          .WORD   CDELET-CBASE2           ;DELETE
001102' 001410                          .WORD   CMODIF-CBASE2           ;MODIFY
001104' 002322                          .WORD   CREPOR-CBASE2           ;REPORT
001106' 001014                          .WORD   CENTER-CBASE2           ;ENTER
001110' 003544                          .WORD   CFETCH-CBASE2           ;/FETCH
001112' 001452                          .WORD   CCONT-CBASE2            ;CONT
001114' 002434                          .WORD   CFILL-CBASE2            ;FILL
001116' 001612                          .WORD   COPSW-CBASE2            ;O/SW
001120' 003466                          .WORD   CSAVE-CBASE2            ;/SAVE
001122' 001436                          .WORD   CSTOP-CBASE2            ;STOP
001124' 005110                          .WORD   CBOC-CBASE2             ;BOC
001126' 004672                          .WORD   CRDM-CBASE2             ;RDM
001130' 001734                          .WORD   CRUN-CBASE2             ;RUN
001132' 004634                          .WORD   CWRM-CBASE2             ;WRM
001134' 003230                          .WORD   CMAP-CBASE2             ;MM
001136' 001332                          .WORD   CENTPB-CBASE2           ;FM
001140' 005016                          .WORD   CADD-CBASE2             ;ADD
001142' 005062                          .WORD   CSUB-CBASE2             ;SUB
```

```
                                        .WORD   CCDB-CBASE2             ;CDB
                                        .WORD   CCBD-CBASE2             ;CBD
                                        .WORD   CKILL-CBASE2            ;KILL
                                        .WORD   CSLDEL-CBASE2           ;/DELETE
                                        .WORD   CLST-CBASE2             ;/LIST
                                        .WORD   CZERO-CBASE2            ;/ZERO
                                        .WORD   CBOOT-CBASE2            ;/BOOT
                                        .IF DF MM
                                        .WORD   CSHIFT-CBASE2           ;SHIFT
                                        .WORD   CUBMAP-CBASE2           ;UBMAP
                                        .ENDC


                                       ;VALID COMMAND TABLE

                                              ;FORMAT OF CODE BYTES

                                              ;FIRST CODE BYTE:

                                              ;  BITS 0 THRU 3 = DISPLACEMENT INTO COM PROC ROUT ADR TBL
                                              ;  BIT 4 = 1     = CMND MAY CONTAIN IMBEDDED PROG # (OPSW)
                                              ;  BIT 5 = 1     = CMND DOES NOT REQUIRE OPERANDS
                                              ;  BIT 6 = 1     = CMND RESTRICTED DURING USER INTERRUPT
                                              ;  BIT 7 = 1     = CMND DOES NOT USE COM PROC ROUT

                                              ;SECOND CODE BYTE:

                                              ;  BITS 0 THRU 6 = DISPLACEMENT INTO CMND ROUT ADR TBL


001162' 000000 044504 050123  CMDTBL: .ASCII  <00><00>'DISPLAY'       ;7 CHARACTER CMNDS
001170' 040514   131
001173'    200 027456 042504          .ASCII  <200><56>'/DELETE'
001200' 042514 042524
001204'    377                        .BYTE   377
001205'    004 040402 051523          .ASCII  <04><02>'ASSIGN'        ;6 CHARACTER CMNDS
001212' 043511   116
001215'    100 042004 046105          .ASCII  <100><04>'DELETE'
001222' 052105   105
001225'    100 046406 042117          .ASCII  <100><06>'MODIFY'
001232' 043111   131
001235'    000 051010 050105          .ASCII  <00><10>'REPORT'
001242' 051117   124
001245'    104 027414 042506          .ASCII  <104><14>'/FETCH'
001252' 041524   110
001255'    377                        .BYTE   377
001256' 005100 047105 042524          .ASCII  <100><12>'ENTER'        ;5 CHARACTER CMNDS
001264'    122
001265'    240 027460 044514          .ASCII  <240><60>'/LIST'
001272' 052123
001274' 012000 051457 053101          .ASCII  <00><24>'/SAVE'
001302'    105
001303'    240 027462 042532          .ASCII  <240><62>'/ZERO'
001310' 047522
001312' 032240 041057 047517          .ASCII  <240><64>'/BOOT'
001320'    124
```

```
                        001                             .IF DF MM
                                                        .ASCII  <100><66>'SHIFT'
                                                        .ASCII  <100><70>'UBMAP'
                        000                             .ENDC
001321'                 377                             .BYTE   377
001322' 007000  047503  052116                          .ASCII  <00><16>'CONT'          ;4 CHARACTER CMNDS
001330' 010004  044506  046114                          .ASCII  <04><20>'FILL'
001336' 011220  050117  053523                          .ASCII  <220><22>'OPSW'
001344' 013000  052123  050117                          .ASCII  <00><26>'STOP'
001352' 026000  044513  046114                          .ASCII  <00><54>'KILL'
001360'                 377                             .BYTE   377
001361'                 002     041030  041517          .ASCII  <02><30>'BOC'           ;3 CHARACTER CMNDS
001366' 015002  042122          115                     .ASCII  <02><32>'RDM'
001373'                 046     051034  047125          .ASCII  <46><34>'RUN'
001400' 017002  051127          115                     .ASCII  <02><36>'WRM'
001405'                 200     040444  042104          .ASCII  <200><44>'ADD'
001412' 023200  052523          102                     .ASCII  <200><46>'SUB'
001417'                 200     041450  041104          .ASCII  <200><50>'CDB'
001424' 025200  041103          104                     .ASCII  <200><52>'CBD'
001431'                 377                             .BYTE   377
001432' 020240  046515                                  .ASCII  <240><40>'MM'           ;2 CHARACTER CMNDS
001436' 021340  046506                                  .ASCII  <340><42>'FM'
001442'                 377                             .BYTE   377                     ;TABLE TERMINATOR
                        001444'                         .EVEN


                                                        ;CONTROL ROUTINE CONSTANTS AND WORK AREAS


001444' 000000                  CRFLGW: .WORD   0                       ;CONTROL ROUTINE FLAGWORD
                        020000          CVLFLG= 20000
                        040000          CFTCHF= 40000
                        100000          CDDFP=  100000

                        001             .IF NDF MM
001446'                 CVMST:
                        000             .ENDC
001446' 000000          CFMST:  .WORD   XXXX                    ;*;START OF FREE MEMORY ADR
                        001             .IF NDF MM
001450'                 CVMEND:
                        000             .ENDC
001450' 000000          CFMEND: .WORD   XXXX                    ;*;END OF FREE MEMORY ADR
001452' 000000          CRMEND: .WORD   XXXX                    ;*;END OF REAL MEMORY
001454' 000000          CPAREA: .WORD   XXXX                    ;*;START ADR OF USER PROGRAM AREA
                        001             .IF DF MM
                        CVMST:  .WORD   P4CONS
                        CVMEND: .WORD   P4CONS+24574.
                        000             .ENDC

001456' 000000          CURCTE: .WORD   0                       ;ADR OF CURRENT CNTRL ROUT TBL ENTRY
001460' 000000          CURPTA: .WORD   0                       ;ADR OF CURRENT PROGRAM TABLE
                        001             .IF DF MM
                        CURPTH: .WORD   0                       ;*;HIGH 16 BITS OF CUR PROG TBL ADR
                        000             .ENDC

001462' 001464'         CPNTR:  .WORD   CCMDRD                  ;CURRENT CMND RD DATA POINTER
```

```
                                CCMDRD: .BLKB   110                     ;CONTROL ROUTINE CONS TERM READIN AREA
                                        .BYTE   015,012
                                CDSTAD: .WORD   0                       ;START ADR OF WORD IN CMND DATA
                                CDCNT:  .WORD   0                       ;NUMBER OF CHARS IN WORD FOUND
                                CERRET: .WORD   0                       ;PRESTORED RETURN ADR FOR ERROR ROUT
                                CR1ADR: .WORD   0                       ;ADR OF COM PROC ROUT FOR CURR CMND
                                CR2ADR: .WORD   0                       ;ADR OF CMND ROUTINE FOR CURR CMND
                                XXXX=   0                               ;DATA THAT WILL BE TAILORED BY MPG
                                CIOALT= 256.                            ;PROG I/O AREA LENGTH DEFINITION
                                CWA:    .BLKW   3                       ;3 WORD WORK AREA
                                CDREND: .WORD   0                       ;END OF DEVICE ROUTINE ADR
                                CMCDVN: .BLKW   10.                     ;MODEL CODE & DEV #'S STORAGE AREA
                                CDFNAM: .ASCII  /ORPHAN/                ;DEFAULT PROGRAM NAME
                                        .IF DF MM
                                MR3VAL: .WORD   0                       ;MMMR3 VALUE - SET BY 1 TIME HSKP
                                MAPMAP: .WORD   0                       ;BIT MAP FOR UNIBUS MAP
                                                                        ;                   28  25
                                                                        ;               30  27  24
                                                                        ;               29  26  23
                                        .WORD   0                       ;REGISTER USAGE (REGS 30 - 7)
                                                                        ;       21  18  15  12  9
                                                                        ;       20  17  14  11  8
                                                                        ;   22  19  16  13  10  7
                                        .ENDC
                                PATCH:  .REPT   10.     ;******* PATCH AREA *******
                                        .WORD   0
                                        .ENDR
```

```
                                .SBTTL  COMMON PROCESSING ROUTINES FOR COMMANDS
                              ;**********
                              ;
                              ;  THIS ROUTINE PROVIDES COMMON PROCESSING AND CHECKS FOR
                              ;  THE FOLLOWING COMMANDS WHICH HAVE A PROGRAM NUMBER
                              ;  AS THE FIRST OPERAND:
                              ;
                              ;       DISPLAY         CONT
                              ;       DELETE          /SAVE
                              ;       MODIFY          STOP
                              ;       REPORT          RUN (CCFMT4)
                              ;       ENTER           KILL
                    001       ;       .IF DF MM
                              ;       SHIFT           MM
                    000       ;       .ENDC
                              ;
                              ;  UPON BRANCH TO CMND ROUTINE:
                              ;
                              ;       R3 = CURRENT PROG TABLE ADR
                              ;       R4 = CURRENT CONTROL ROUT TBL ENTRY ADR
                              ;
                              ;  ALSO PRESENT ARE END OF COMMAND CHECKS WHICH MAY BE
                              ;  USED BY ALL COMMANDS
                              ;
                              ;**********

001676' 005767  006312  CCFMT4: TST     CSCFWD                  ;PROG # PRESENT?
001702' 100002          CCFMT1: BPL     CCFMT1                  ;N,Y-CCFMT1
001704' 000167  176560          JMP     CGORUN                  ;GO INITIATE PROG EXECUTION

001710' 004567  006570  CCFMT1: JSR     R5,CVPNUM               ;GO VALIDATE PROG # IN CMND DATA
                    001         .IF DF MM
                                JSR     PC,UPAPAR               ;MAP PAR'S TO USER'S PROG AREA
                    000         .ENDC
001714' 010700                  MOV     PC,R0                   ;SET UP ADR OF ENTER CMND ROUT
001716' 062700  000122          ADD     #CENTER-.,R0
001722' 020067  177660          CMP     R0,CR2ADR               ;IS THIS THE ENTER CMND?
001726' 001402                  BEQ     CCFMTJ                  ;N,Y-CCFMTJ
001730' 005703                  TST     R3                      ;PROG EXIST FOR THIS #?
001732' 001402                  BEQ     CCFPDE                  ;Y,N-CCFPDE
001734' 000177  177646  CCFMTJ: JMP     @CR2ADR                 ;GO TO ROUTINE FOR THIS CMND

001740' 004567  004260  CCFPDE: JSR     R5,CTYPE                ;GO REPORT PROG DOES NOT EXIST ERROR
001744' 000006                  .WORD   006
001746' 004677                  .WORD   CERM05-.

                                ;  COMMON COMMAND RETURN POINTS

001750' 005767  006240  CCFMTA: TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
001754' 100355          CCFMT1: BPL     CCFMT1                  ;N,Y-CCFMT1
001756' 000167  176404  CCFRET: JMP     CNCMND                  ;GO REQUEST NEXT CMND
001762' 005767  006226  CCFMTB: TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
001766' 100773          CCFRET: BMI     CCFRET                  ;Y,N-CCFRET
001770' 004567  006226          JSR     R5,CSCN                 ;GO SCAN IT
```

F02


```
   880  001774' 004567  004224                    JSR     R5,CTYPE            ;GO REPORT UNNECESSARY OPERAND ERROR
   881  002000' 000102                             .WORD   102
   882  002002' 004661                             .WORD   CERM06-.
```

```
                                ;**********
                                ;
                                ;  THIS ROUTINE PROVIDES COMMON PROCESSING FOR THE FOLLOWING COMMANDS
                                ;  WHICH HAVE A WORD ADDRESS IN OCTAL AS THE FIRST OPERAND:
                                ;
                                ;       BOC
                                ;       RDM
                                ;       WRM
                                ;
                                        .IF DF MM
                                        .IFF
                                ;  R4 CONTAINS CONVERTED BINARY WORD UPON BRANCH TO CMND ROUTINE
                                        .IFT
                                ;  R3 & R4 CONTAINS CONVERTED BINARY WORDS UPON BRANCH TO CMND ROUTINE
                                        .ENDC
                                ;
                                ;**********

002004' 004567  006224          CCFMT2: JSR     R5,CSCNB8               ;SCAN OPERAND & CONVERT TO BINARY
002010' 004567  011330                  JSR     R5,CKODAD               ;GO CHECK FOR ODD ADR
002014' 010004                          MOV     R0,R4                   ;SAVE 16 BIT ADR IN R4
                                        .IF DF MM
                                        MOV     R2,R3                   ;SAVE HIGH 6 BITS IN R3
                                        .ENDC
002016' 000746                          BR      CCFMTJ                  ;GO TO ROUTINE FOR THIS CMND


                                ;**********
                                ;
                                ;  THIS ROUTINE PROVIDES COMMON PROCESSING FOR THE FOLLOWING
                                ;  COMMANDS WHICH HAVE AN ASCII WORD AS THE FIRST OPERAND:
                                ;
                                ;       ASSIGN
                                ;       FETCH
                                ;       FILL
                                ;
                                ;**********

002020' 004567  006176          CCFMT3: JSR     R5,CSCN                 ;SCAN FIRST OPERAND
002024' 005702                          TST     R2                      ;ANOTHER OPERAND PRESENT?
002026' 100342                          BPL     CCFMTJ                  ;N,Y-CCFMTJ
002030' 004567  004170          CN2OER: JSR     R5,CTYPE                ;REPORT NO 2ND OPERAND ERROR
002034' 000042                          .WORD   042
002036' 004643                          .WORD   CERMD8-.
```

```
                                        .SBTTL  ENTER, DELETE, & MODIFY COMMAND ROUTINES


                                ;       ********************
                                ;       *                  *
                                ;       *  ENTER COMMAND   *
                                ;       *                  *
                                ;       ********************


002040' 042767  040000  177376  CENTER: BIC     #CFTCHF,CRFLGW          ;RESET THE FETCH FLAG
002046' 010700                          MOV     PC,R0                   ;SET ADR OF PROG'S NAME TO
002050' 062700  177574                  ADD     #CDFNAM-.,R0            ;THE DEFAULT NAME
002054' 005005                          CLR     R5                      ;RESET "SAVE" FLAG
002056' 000167  002440                  JMP     CSVCOM                  ;GO PROCESS PROG NAME
            001                         .IF DF MM
                                        .IFT
                                CENTPA: JSR     R5,CTYPE                ;ASK FOR PROG AREA ADR
                                        .WORD   420
                                        .WORD   ADRREQ-.
                                        .WORD   17.
                                        JSR     R5,@CTRD                ;ISSUE READ FOR THE REPLY
                                        .WORD   CCMDRD-.
                                        .WORD   71.
                                        MOV     PC,R0                   ;SET COMMON ERROR RET ADR
                                        ADD     #CENTPA-.,R0            ;TO ISSUE OF 'PROG AREA ADR' MSG
                                        MOV     R0,CERRET
                                        TST     (R4)                    ;PROG ALREADY EXIST IN THIS SLOT?
                                        BEQ     2$                      ;Y,N-2$
                                        JSR     PC,MAPHKP               ;CK & DEALLOCATE UNIBUS MAP REGS
                                        CLR     (R4)                    ;RESET PROG'S ADR IN SLOT TBL
                                        TST     CRTEXT(R4)              ;WAS IT A FIXED ADR PROG?
                                        BMI     1$                      ;N,Y-1$
                                        JSR     R5,CSDNRC               ;SHIFT DOWN MPG CONTROLLED PROGS
                                1$:     CLR     CRTEXT(R4)              ;HSKP SLOT'S EXTENSION WORD
                                2$:     CLR     CENADR                  ;RESET PROG AREA ADR
                                        JSR     R5,CSCNSP               ;SCAN FOR THE REPLY
                                        BIT     #CSCFND,R2              ;WAS ANY DATA ENTERED?
                                        BEQ     CENPAA                  ;Y,N-CENPAA
                                        JSR     R5,ASOCBN               ;CONVERT THE REPLY TO BINARY
                                        .WORD   CENERB-.
                                        JSR     PC,CKPGAD               ;GO CK FOR VALID PROG ADR
                                        MOV     R2,CENADR               ;SAVE HI 16 BITS OF ADR
                                        CMP     R2,HIMPGP               ;ADR WITHIN MPG CONTROLLED PROGS?
                                        BHIS    3$                      ;Y,N-3$
                                        JMP     SHPGE2                  ;ISSUE 'PROG AREA IN USE' ERR MSG
                                3$:     MOV     R2,R1                   ;PROG ADR TO S/R'S REG
                                        JSR     PC,CKF3PG               ;CK IF AREA IS OCCUPIED
                                        TST     R2                      ;WAS AREA OCCUPIED?
                                        BEQ     CENTPA                  ;N,Y-CENTPA
                                CENPAA: JSR     R5,CTYPE                ;ISSUE 'ASGN DEV:' MSG
                                        .IFF
002062' 004567  004136          CENTPA: JSR     R5,CTYPE                ;ISSUE 'ASGN DEV' MSG
            000                         .ENDC
002066' 000420                          .WORD   420
002070' 005443                          .WORD   CADMSG-.
```

```
 985  002072' 000013                          .WORD   11.
 986  002074' 004577  175730                  JSR     R5,@CTRD                ;ISSUE RD FOR REPLY
 987  002100' 177364                          .WORD   CCMDRD-.
 988  002102' 000107                          .WORD   71.
 989  002104' 010700                          MOV     PC,R0                   ;SET COMMON ERROR RET ADR
 990             001                          .IF DF MM
 991                                          .IFF
 992  002106' 062700  177754                  ADD     #CENTPA-.,R0            ;TO ISSUE OF 'ASGN DEV:' MSG
 993                                          .IFT
 994                                          ADD     #CENPAA-.,R0            ;TO ISSUE OF 'ASGN DEV:' MSG
 995             000                          .ENDC
 996  002112' 010067  177464                  MOV     R0,CERRET
 997             001                          .IF DF MM
 998                                          MOV     #P4CONS,CVMST           ;INIT VIRTUAL MEM START ADR
 999             000                          .ENDC
1000  002116' 004567  010446                  JSR     R5,CVLMD2               ;GO VALIDATE MODEL & DEV #'S
1001  002122' 122767  000377  177470          CMPB    #377,CMCDVN             ;THIS DEFAULT TYPE OF DEV?
1002  002130' 001006                          BNE     5$                      ;Y,N-5$
1003  002132' 126767  040441  177461          CMPB    CNONID+5,CMCDVN+1       ;IS IT "NONE"?
1004  002140' 001402                          BEQ     5$                      ;N,Y-5$
1005  002142' 000167  003472                  JMP     CASER1                  ;GO REPORT INV DEV TYPE
1006  002146' 004767  011226          5$:     JSR     PC,CSUERR               ;SET UP ERROR RETURN ADR
1007             001                          .IF DF MM
1008                                          TST     CENADR                  ;WAS AN ADR SPECIFIED FOR THIS PROG?
1009                                          BEQ     7$                      ;Y,N-7$
1010                                          BIS     #FIXADR,CRTEXT(R4)      ;SET ITS FIXED ADR FLAG
1011             000                          .ENDC
1012  002152' 004567  006416          7$:     JSR     R5,CSPGUP               ;SHIFT UP HIGHER PROGRAMS IF NEEDED
1013             001                          .IF DF MM
1014                                          .IFT
1015                                          MOV     CENADR,R3               ;GET ENTERED PROG AREA ADR
1016                                          BNE     9$                      ;WAS THERE ONE? (N,Y-9$)
1017                                          MOV     CURPTH,R3               ;GET START ADR FOR THIS PROG
1018                                  9$:     JSR     PC,UPAPAR               ;MAP KERNEL PAR'S TO USR PROG AREA
1019                                          .IFF
1020  002156' 016703  177276                  MOV     CURPTA,R3               ;GET START ADR FOR THIS PROG
1021             000                          .ENDC
1022  002162' 010300                          MOV     R3,R0                   ;SET UP END OF PROG TBL WORK ADR
1023  002164' 062700  000242                  ADD     #PTLGTH,R0
1024  002170' 020067  177254                  CMP     R0,CVMEND               ;ENOUGH ROOM FOR PROG TBL?
1025  002174' 103102                          BHIS    CENERR                  ;Y,N-CENERR
1026  002176' 012740  000242                  MOV     #PTLGTH,-(R0)           ;STORE PROG TBL LENGTH WD
1027  002202' 012701  000120                  MOV     #PTLGTH-2/2,R1          ;GET # OF WORDS TO CLEAR
1028  002206' 005040                  10$:    CLR     -(R0)                   ;CLEAR WORD OF PROG TBL
1029  002210' 005301                          DEC     R1                      ;DECR WORD CNT
1030  002212' 001375                          BNE     10$                     ;CNT=0? (Y,N-10$)
1031  002214' 012763  100000  000002          MOV     #100000,POPSW(R3)       ;INITIALIZE OPSW WORD
1032  002222' 004767  011362                  JSR     PC,CPNASC               ;GET PROG # IN ASCII
1033  002226' 010163  000006                  MOV     R1,PASCIN(R3)           ;STORE IT IN PROG TBL
1034  002232' 010301                          MOV     R3,R1                   ;GET ADR OF PROG NAME IN PROG TBL
1035  002234' 062701  000010                  ADD     #PNAME,R1
1036  002240' 010700                          MOV     PC,R0                   ;GET ADR OF STORED NAME
1037  002242' 062700  175536                  ADD     #CLOCZ-.,R0
1038  002246' 012021                          MOV     (R0)+,(R1)+             ;MOVE NAME TO PROG TBL
1039  002250' 012021                          MOV     (R0)+,(R1)+
1040  002252' 012021                          MOV     (R0)+,(R1)+
```

```
1041  002254' 004567 011134          JSR     R5,CSTDVI       ;GO STORE DEV INFO IN PROG TBL
1042  002260' 004567 007356          JSR     R5,CLDEVR       ;GO LOAD THIS DEV ROUT
1043  002264' 000116                 .WORD   CENERR-.        ;    INSUF MEM ERR RET ADR
1044  002266' 000110                 .WORD   CENDER-.        ;    DEV ERROR ERR RET ADR
1045  002270' 032767 040000 177146   BIT     #CFTCHF,CRFLGW  ;'FETCH' FLAG SET?
1046  002276' 001417                 BEQ     30$             ;Y,N-30$
1047  002300' 010300                 MOV     R3,R0           ;GET ADR OF SOURCE FILE NAME
1048  002302' 062700 000010          ADD     #PNAME,R0
1049  002306' 010701                 MOV     PC,R1           ;GET EXEC INTERFACE STORAGE ADR
1050  002310' 062701 175470          ADD     #CLOCZ-.,R1
1051  002314' 012021                 MOV     (R0)+,(R1)+     ;STORE PROG'S NAME THERE
1052  002316' 012021                 MOV     (R0)+,(R1)+
1053  002320' 011011                 MOV     (R0),(R1)
1054  002322' 004577 175526          JSR     R5,@OPENL       ;GO OPEN SOURCE FILE
1055             001                 .IF DF MM
1056                                 .IFF
1057  002326' 000050                 .WORD   CENDER-.
1058                                 .IFT
1059                                 .WORD   CENRSM-.
1060             000                 .ENDC
1061  002330' 004567 011400          JSR     R5,FETCH        ;GO TO FETCH & COMPILE PROG ROUT
1062  002334' 000406                 BR      CENTPC          ;GO CK FOR SHIFT DOWN
1063  002336' 004567 003662    30$:  JSR     R5,CTYPE        ;ISSUE "ENTER STMNT'S" MSG
1064  002342' 000600                 .WORD   600
1065  002344' 005367                 .WORD   CESMSG-.
1066  002346' 004567 011266          JSR     R5,ENTER        ;GO TO ENTER & COMPILE PROG ROUT
1067  002352' 010067 177070  CENTPC: MOV     R0,CVMST        ;STORE NEW FREE MEM START ADR
1068             001                 .IF DF MM
1069                                 MOV     #P4CONS,R3      ;RESTORE PROG TBL ADR
1070                                 BIC     R3,PRDIOA(R3)   ;ADJ PROG TBL ADR'S TO
1071                                 BIC     R3,PWRIOA(R3)   ;USER PAGES
1072                                 BIC     R3,PSRCST(R3)
1073                                 BIC     R3,POBJST(R3)
1074                                 MOV     PLNGTH(R3),R0   ;GET PROG'S LENGTH
1075                                 BIT     #77,R0          ;PROG END AT 32 WD BNDRY?
1076                                 BEQ     80$             ;N,Y-80$
1077                                 ADD     #100,R0         ;ADJ IT TO NXT HIGHEST 32 WD BNDRY
1078                           80$:  MOV     #6,R1           ;SET UP LOOP CNT
1079                           90$:  ASR     R0              ;CONVERT IT TO A CNT OF 32 WD
1080                                 DEC     R1              ;BLOCKS
1081                                 BNE     90$
1082                                 MOV     CURCTE,R4       ;RESTORE CNTL ROUT TBL ADR
1083                                 BIS     R0,CRTEXT(R4)   ;STORE LGTH IN CNTRL ROUT TBL EXT.
1084             000                 .ENDC
1085  002356' 004567 006406  CENTPB: JSR     R5,CSDNRC       ;SHIFT DOWN PROG'S
1086  002362' 004567 003636          JSR     R5,CTYPE        ;ISSUE PGMS COMP/MEM FMTTED MSG
1087  002366' 000400                 .WORD   400
1088  002370' 005363                 .WORD   CPCMRF-.
1089  002372' 000167 175770          JMP     CNCMND          ;RETURN FOR NEXT CMND
1090
1091             001                 .IF DF MM
1092                           CENADR: .WORD   0               ;*;PROG ADR HOLD
1093
1094                           CENRSM: JSR     PC,MAPHKP       ;HSKP MAPMAP WDS IF NEEDED
1095             000                 .ENDC
1096  002376' 005014           CENDER: CLR     (R4)            ;RESET PROG'S TBL ENTRY
```

```
1097  002400' 000766                          BR      CENTPB          ;GO SHIFT ANY PROGS DOWN
1098
1099  002402' 005014                  CENERR: CLR     (R4)            ;RESET THIS PROG'S CTRL ROUT TBL ADR
1100  002404' 004567  003614          CENERA: JSR     R5,CTYPE        ;ISSUE INSUFF MEM MSG
1101  002410' 000006                          .WORD   006
1102  002412' 004455                          .WORD   CERM23-.
1103             001                          .IF DF MM
1104
1105                                  CENERB: JSR     R5,CTYPE        ;ISSUE DATA CONV
1106                                          .WORD   302             ;ERROR MSG
1107                                          .WORD   CERM02-.
1108             000                          .ENDC
1109
1110
1111                                  ;       **********************
1112                                  ;       *                    *
1113                                  ;       *  DELETE COMMAND    *
1114                                  ;       *                    *
1115                                  ;       **********************
1116
1117
1118  002414' 004767  010760          CDELET: JSR     PC,CSUERR       ;GO SET UP ERROR RETURN ADR
1119             001                          .IF DF MM
1120                                          JSR     PC,MAPHKP       ;CK & DEALLOCATE UNIBUS MAP REGS
1121             000                          .ENDC
1122  002420' 005014                          CLR     (R4)            ;CLEAR THIS PROG'S CTRL ROUT TBL ENTRY
1123  002422' 005767  005566                  TST     CSCFWD          ;ANOTHER PROG # IN CMND
1124  002426' 100753                          BMI     CENTPB          ;Y,N-CENTPB
1125  002430' 000167  177254                  JMP     CCFMT1          ;GO SCAN NXT PROG #
```

```
                                ;       ********************
                                ;       *                  *
                                ;       *  MODIFY COMMAND  *
                                ;       *                  *
                                ;       ********************

002434' 004767  010740          CMODIF: JSR     PC,CSUERR               ;SET UP ERROR RETURN ADR
            001                         .IF DF MM
                                        .IFT
                                        MOV     (R4),CENADR             ;STORE HI 16 BITS OF PROG AREA ADR
                                        MOV     R3,CVMST                ;INIT VIRTUAL MEM START ADR
                                        BIS     R3,PRDIOA(R3)           ;ADJ PROG TBL ADR'S TO
                                        BIS     R3,PWRIOA(R3)           ;PAGE 4
                                        BIS     R3,PSRCST(R3)
                                        BIS     R3,POBJST(R3)
                                        .IFTF
002440' 004567  006130          10$:    JSR     R5,CSPGUP               ;GO SHIFT UP HIGHER PROGRAMS
                                        .IFT
                                        MOV     (R4),R3                 ;GET PROG AREA ADR
                                        JSR     PC,UPAPAR               ;MAP KERNEL'S PAR'S TO USR PRG AREA
            000                         .ENDC
002444' 004567  003554                  JSR     R5,CTYPE                ;ISSUE "ENTER STMNT'S" MSG
002450' 000600                          .WORD   600
002452' 005261                          .WORD   CESMSG-.
002454' 004567  011242                  JSR     R5,MODIFY               ;GO PROCESS CHANGES & RECOMPILE
002460' 000734                          BR      CENTPC                  ;GO SHIFT DOWN PROG'S
            001                         .IF DF MM
                                        .PAGE
                                        .SBTTL  SHIFT AND UBMAP COMMANDS


                                ;       ********************
                                ;       *                  *
                                ;       *  SHIFT COMMAND   *
                                ;       *                  *
                                ;       ********************


                                CSHIFT: TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
                                        BPL     10$                     ;N,Y-10$
                                        JMP     CN2OER                  ;GO REPORT NO 2ND OPER ERROR
                                10$:    CMPB    #'T,@CPNTR              ;IS ITS FIRST CHAR A "T"?
                                        BEQ     20$                     ;N,Y-20$
                                        JSR     R5,CTYPE                ;ISSUE 'TO' NOT SPECIFIED ERROR
                                        .WORD   042                     ;MSG
                                        .WORD   CERM36-.
                                20$:    JSR     R5,CSCN                 ;SCAN OVER THE 'TO'
                                        TST     R2                      ;DESTINATION OPERAND SUPPLIED?
                                        BPL     30$                     ;N,Y-30$
                                        JSR     R5,CTYPE                ;ISSUE NO DESTINATION ERROR MSG
                                        .WORD   042
                                        .WORD   CERM37-.
                                30$:    MOV     CPNTR,-(SP)             ;SAVE NXT OPERAND PNTR
                                        JSR     R5,CSCN                 ;SCAN DESTINATION OPERAND
                                        MOV     (SP)+,CPNTR             ;RESTORE PNTR IN CASE OF A PROG ADR
```

```
1183                                    CMPB    #'M,@CDSTAD             ;IS ITS FIRST CHAR AN 'M'?
1184                                    BEQ     SHTMPG                  ;N,Y-SHTMPG
1185                                    JSR     R5,ASDECD               ;CONVERT DATA TO PACKED DECIMAL
1186                                    .WORD   SHPGNA-.
1187                                    JSR     R5,DECBIN               ;CONVERT PKED DEC TO BINARY
1188                                    .WORD   SHPGNA-.
1189                                    CMP     R0,#MAXPRG              ;# LARGER THAN HIGHEST PROG #?
1190                                    BHI     SHPGNA                  ;N,Y-SHPGNA
1191
1192                                    ;SHIFT PROGRAM TO DIFFERENT SLOT
1193
1194                            SHMVSL: MOV     R4,R5                   ;SAVE ORIG SLOT ADR
1195                                    JSR     R5,CVPNM1               ;GO VALIDATE DEST OPER AS A PROG #
1196                                    CMP     R4,R5                   ;THE YO-YO SPECIFY SAME SLOT #'S?
1197                                    BEQ     SHPGND                  ;N,Y-SHPGND
1198                                    TST     CRTEXT(R5)              ;IS ORIG PROG AT A FIXED ADR?
1199                                    BMI     SHMVFA                  ;N,Y-SHMVFA
1200                                    JSR     PC,CSUERR               ;SET UP ERROR RET ADR
1201                                    JSR     R5,CSPGUP               ;SHIFT UP ANY PROGS ABOVE NEW SLOT
1202                                    MOV     (R4),R1                 ;SET UP OPEN AREA ADR
1203                                    JSR     PC,CKF3PG               ;GET SIZE OF AVAIL MEM IN R2
1204                                    MOV     CRTEXT(R5),R3           ;GET PROG'S LENGTH
1205                                    CMP     R2,R3                   ;ENOUGH ROOM FOR PROG?
1206                                    BLO     SHPGE1                  ;Y,N-SHPGE1
1207                                    MOV     (R4),-(SP)              ;SAVE ITS NEW ADR
1208                                    MOV     R5,-(SP)                ;SAVE ORIG SLOT ADR
1209                                    MOV     (R5),(R4)               ;SET UP 'FROM' ADR
1210                                    MOV     R1,R5                   ;SET UP DEST. ADR
1211                                    JSR     PC,MOVPGD               ;MOVE PROG TO NEW MPG ADR
1212                                    MOV     (SP)+,R5                ;RESTORE ORIG SLOT ADR
1213                                    MOV     (SP)+,(R5)              ;STORE NEW PROG ADR
1214                            SHMVFA: MOV     (R5),(R4)               ;MOVE PROG ADR TO NEW SLOT
1215                                    MOV     CRTEXT(R5),CRTEXT(R4)   ;MOVE ITS EXTENSION WORD ALSO
1216                                    CLR     (R5)                    ;RESET OLD PROG SLOT ENTRY
1217                                    MOV     (R4),R3                 ;GET ITS START ADR
1218                                    JSR     PC,UPAPAR               ;MAP PAR'S TO USER'S AREA
1219                                    JSR     PC,CPNASC               ;GET PROG'S NEW SLOT # IN ASCII
1220                                    MOV     R1,PASCIN(R3)           ;STORE IT IN PROG TBL
1221                            SHPGEX: JMP     CENTPB                  ;GO RECOMPILE PROG'S AND EXIT
1222                            SHPGND: JMP     CNCMND                  ;RETURN - NO PROGS MOVED
1223
1224                                    ;SHIFT PROGRAM TO MPG CONTROL OF ITS ADDRESS
1225
1226                            SHTMPG: TST     CRTEXT(R4)              ;IS IT ALREADY UNDER MPG CONTROL?
1227                                    BPL     SHPGND                  ;N,Y-SHPGND
1228                                    JSR     PC,CSUERR               ;SET UP ERROR RET ADR
1229                                    MOV     R4,R0                   ;SET PROG SLOT ADR IN R0
1230                                    JSR     R5,CSPNFP               ;SHIFT UP ANY MPG CONTROLLED PROGS
1231                                    MOV     CSPNPA,R1               ;SET UP OPEN AREA START ADR
1232                                    MOV     R1,R5                   ;SAVE IT FOR MOVE S/R
1233                                    JSR     PC,CKF3PG               ;GET SIZE OF AVAIL AREA IN R2
1234                                    MOV     CRTEXT(R4),R3           ;GET PROG'S LENGTH
1235                                    BIC     #FIXADR,R3              ;RESET ITS FIXED ADR FLAG
1236                                    CMP     R2,R3                   ;ENOUGH ROOM FOR PROG?
1237                                    BLO     SHPGE1                  ;Y,N-SHPGE1
1238                                    JSR     PC,MOVPGD               ;MOVE PROG INTO MPG'S AREA
```

```
1239                                    MOV     CSPNPA,(R4)             ;STORE ITS NEW ADR IN SLOT TBL
1240                                    BIC     #FIXADR,CRTEXT(R4)      ;RESET ITS FIXED ADR FLAG
1241                                    BR      SHPGEX                  ;GO TO CMND EXIT
1242
1243                                    ;SHIFT PROGRAM TO NEW FIXED ADDRESS
1244
1245                            SHPGNA: JSR     R5,CSCN                 ;RE-SCAN THE DEST OPERAND
1246                                    JSR     R5,ASOCBN               ;CONVERT ASCII DATA TO BINARY
1247                                    .WORD   CENERB-.
1248                                    JSR     PC,CKPGAD               ;CK IF PROG ADR IS VALID
1249                                    CMP     R2,(R4)                 ;NEW ADR SAME AS OLD?
1250                                    BEQ     SHPGND                  ;N,Y-SHPGND
1251                                    CMP     R2,HIMPGP               ;ADR WITHIN MPG CONTROLLED PROGS?
1252                                    BLO     SHPGE2                  ;N,Y-SHPGE2
1253                                    MOV     R2,R5                   ;SAVE SUPPLIED ADR
1254                                    MOV     R2,R1                   ;SET IT IN S/R'S REG
1255                                    JSR     PC,CKF3PG               ;CHECK TO SEE HOW MUCH MEM AVAIL
1256                                    TST     R2                      ;ANY ROOM FOR PROG?
1257                                    BEQ     SHPGND                  ;Y,N-SHPGND
1258                                    MOV     CRTEXT(R4),R3           ;GET PROG'S LGTH IN 32 WD BLKS
1259                                    BIC     #FIXADR,R3              ;RESET FIXED ADR FLG IF THERE
1260                                    CMP     R2,R3                   ;ENOUGH ROOM FOR THIS PROG?
1261                                    BLO     SHPGE1                  ;Y,N-SHPGE1
1262                                    MOV     R5,-(SP)                ;SAVE ITS STARTING ADR
1263                                    JSR     PC,MOVPGD               ;MOV PROG TO ITS NEW AREA
1264                                    MOV     (SP)+,(R4)              ;STORE ITS START ADR
1265                                    BIS     #FIXADR,CRTEXT(R4)      ;MAKE IT A FIXED ADR PROGRAM
1266                                    MOV     (R4),R3                 ;SET UP ITS BASE ADR
1267                                    JSR     PC,UPAPAR               ;MAP PAR'S TO THE NEW AREA
1268                                    BIT     #UNIMAP,CSYSFW          ;USING UNIBUS MAP?
1269                                    BNE     SHPGEX                  ;N,Y-SHPGEX
1270                                    BIT     #USEUBM,(R3)            ;PROG USE UNIBUS MAP?
1271                                    BEQ     SHPGEX                  ;Y,N-SHPGEX
1272                                    JSR     PC,CKUBAD               ;CK IF HE'S OUT OF RANGE
1273                                    BR      SHPGEX                  ;GO TO EXIT
1274
1275                            SHPGE1: JSR     R5,CTYPE                ;GO REPORT INSUF. ROOM FOR PROG
1276                                    .WORD   002
1277                                    .WORD   CERM38-.
1278                            SHPGE2: JSR     R5,CTYPE                ;ISSUE 'PROG AREA IN USE'
1279                                    .WORD   002                     ;ERROR MSG
1280                                    .WORD   CERM34-.
1281
1282
1283                                    ;VALIDATE PROGRAM AREA ADR
1284
1285                                            ;JSR    PC,CKPGAD       S/R CALL
1286                                            ;
1287                                            ;R0 & R2 CONTAINS A 22 BIT ADR
1288                                            ;
1289                                            ;ON EXIT: R2 CONTAINS HI 16 BITS OF ADR
1290                                            ;DESTROYS R0,R1
1291
1292                            CKPGAD: MOV     #10.,R1                 ;GET HI 16 BITS OF ADR
1293                            10$:    ROL     R0                      ;IN R2
1294                                    ROL     R2
```

```
1295                                    DEC     R1
1296                                    BNE     10$
1297                                    BIT     #176000,R0              ;ADR ON A 32 WD BNDRY?
1298                                    BEQ     20$                     ;N,Y-20$
1299                                    INC     R2                      ;POINT TO NXT 32 WD BLK
1300                            20$:    CMP     R2,CPAREA               ;IS ADR WITHIN MPG AREA?
1301                                    BHIS    30$                     ;Y,N-30$
1302                                    JSR     R5,CTYPE                ;ISSUE "ADR WITHIN MPG" ERROR
1303                                    .WORD   003                     ;MSG
1304                                    .WORD   CERM35-.
1305                            30$:    CMP     R2,CRMEND               ;IS ADR > MEM?
1306                                    BLOS    40$                     ;Y,N-40$
1307                                    JSR     R5,CTYPE                ;ISSUE "ADR > MEM" ERROR
1308                                    .WORD   003                     ;MSG
1309                                    .WORD   CERM09-.
1310                            40$:    RTS     PC                      ;EXIT IN-LINE
1311
1312
1313                            ;       ********************
1314                            ;       *                  *
1315                            ;       *  UBMAP COMMAND   *
1316                            ;       *                  *
1317                            ;       ********************
1318
1319
1320                            CUBMAP: BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
1321                                    BEQ     SHPGND                  ;Y,N-SHPGND
1322                                    TST     PUBMAP(R3)              ;THIS PROG USE UNIBUS MAP?
1323                                    BEQ     SHPGND                  ;Y,N-SHPGND
1324                                    JSR     PC,MAPHKP               ;RESET REGS CURRENTLY USING
1325                                    CLR     PUBMAP(R3)              ;RESET THIER POINTERS
1326                                    JSR     R5,CRQUBM               ;GO REQUEST NEW MAP REGS
1327                                    NOP                             ;FILL INST
1328                                    NOP                             ;FILL INST
1329                                    BR      SHPGEX                  ;GO RECOMPILE PROG'S AND EXIT
1330                    000             .ENDC
```

```
                                        .SBTTL  STOP, CONT, KILL, & OPSW COMMAND ROUTINES

                                ;       ******************
                                ;       *                *
                                ;       *  STOP COMMAND  *
                                ;       *                *
                                ;       ******************

002462' 005713                  CSTOP:  TST     (R3)                    ;THIS PROG ACTIVE?
002464' 100011                          BPL     CINACT                  ;Y,N-CINACT
002466' 052713  000002                  BIS     #URSTOP,(R3)            ;SET USER STOP FLAG FOR THIS PROG
002472' 000167  177252          CLSJMP: JMP     CCFMTA                  ;GO CHECK FOR ANOTHER PROG #

                                ;       ******************
                                ;       *                *
                                ;       *  CONT COMMAND  *
                                ;       *                *
                                ;       ******************

002476' 005713                  CCONT:  TST     (R3)                    ;THIS PROG ACTIVE?
002500' 100003                          BPL     CINACT                  ;Y,N-CINACT
002502' 042713  000006                  BIC     #URSTOP+ERSTOP,(R3)     ;RESET USER & ERROR STOP FLAGS
002506' 000771                          BR      CLSJMP                  ;EXIT

002510' 004567  003510          CINACT: JSR     R5,CTYPE                ;ISSUE 'INACTIVE' ERROR MSG
002514' 000006                          .WORD   006
002516' 004445                          .WORD   CERM27-.

                                ;       ******************
                                ;       *                *
                                ;       *  KILL COMMAND  *
                                ;       *                *
                                ;       ******************

002520' 005713                  CKILL:  TST     (R3)                    ;THIS PROG ACTIVE?
002522' 100372                          BPL     CINACT                  ;Y,N-CINACT
002524' 042713  100000                  BIC     #ACTIVE,(R3)            ;RESET ITS ACTIVE FLAG
002530' 005767  031460                  TST     CSYSFW                  ;CONSOLE REQ BEING PROCESSED?
002534' 100356                          BPL     CLSJMP                  ;Y,N-CLSJMP
002536' 032713  000040                  BIT     #SETDED,(R3)            ;THIS PROG SET PRT DEV DEDICATED FLG?
002542' 001415                          BEQ     5$                      ;Y,N-5$
002544' 032767  004000  031442          BIT     #CHPINT,CSYSFW          ;IS THE PRINTER BEING USED?
002552' 001404                          BEQ     2$                      ;Y,N-2$
002554' 016700  175234                  MOV     CLISTI,R0               ;GET ADR OF PRINTER INFO
002560' 005070  000010                  CLR     @8.(R0)                 ;RESET PRINTER'S INT ENABLE
002564' 042767  064000  031422  2$:     BIC     #CTUDED+CRUDED+CHPINT,CSYSFW ;RESET PRT DEV DED. FLAGS
002572' 042713  000040          5$:     BIC     #SETDED,(R3)            ;RESET THIS PROG'S "I SET IT" FLG
002576' 021467  032064                  CMP     (R4),CPGCPT             ;IS THIS THE PROG THAT'S RUNNING?
002602' 001005                          BNE     10$                     ;Y,N-10$
                  001                   .IF DF MM
                                        .IFF
002604' 010700                          MOV     PC,R0                   ;SET UP ABORT RET ADR FOR PROG
002606' 062700  032406                  ADD     #CKILEX-.,R0
002612' 010066  000014                  MOV     R0,12.(SP)              ;STORE ADR IN PROG'S PC ON STACK
```

```
1388                                            .IFT
1389                                            MOV     #CKILEX-PG6BA+P6CONS,18.(SP) ;SET UP ABORT RET ADR FOR PROG
1390                                            BIT     #140000,20.(SP)         ;WAS USER IN KERNEL MODE?
1391                                            BNE     10$                     ;Y,N-10$
1392                                            MOV     PC,R0                   ;SET UP KERNEL MODE
1393                                            ADD     #CKILEK-.,R0            ;ABORT RETURN ADR
1394                                            MOV     R0,18.(SP)              ;STORE IT INSTEAD
1395                    000                     .ENDC
1396  002616' 010300                    10$:    MOV     R3,R0                   ;GET ADR OF DEV ROUTINE KILL
1397  002620' 062700  000302                    ADD     #PTLGTH+DKILAD,R0       ;ROUT ADR
1398  002624' 005710                            TST     (R0)                    ;IS THERE A KILL ROUT IN THE
1399  002626' 001721                            BEQ     CLSJMP                  ;DEV ROUT? (Y,N-CLSJMP)
1400  002630' 061000                            ADD     (R0),R0                 ;MAKE ITS ADR ABSOLUTE
1401                    001                     .IF DF MM
1402                                            MOV     SPPAR6,@#KPAR6          ;POINT PAR 6 TO SHARED CODE
1403                    000                     .ENDC
1404  002632' 004510                            JSR     R5,(R0)                 ;GO TO KILL ROUTINE
1405  002634' 000716                            BR      CLSJMP                  ;EXIT
1406
1407
1408                                    ;       ********************
1409                                    ;       *                  *
1410                                    ;       *  OPSW COMMAND    *
1411                                    ;       *                  *
1412                                    ;       ********************
1413
1414  002636' 004567  010630            COPSW:  JSR     R5,CISDNM               ;GO ISOLATE PROG # IN CMND
1415  002642' 005701                            TST     R1                      ;WAS THERE A NUMBER?
1416  002644' 001030                            BNE     COPSWN                  ;N,Y-COPSWN
1417  002646' 004567  005362                    JSR     R5,CSCNCB               ;SCAN AND CONVERT OPSW VALUE
1418  002652' 010705                            MOV     PC,R5                   ;SET UP ADR OF CNTRL ROUT TBL
1419  002654' 062705  031342                    ADD     #CRTBL-.,R5
1420  002660' 005004                            CLR     R4                      ;RESET PROG FND FLAG
1421  002662' 005715                    10$:    TST     (R5)                    ;THIS ENTRY IN USE?
1422  002664' 001404                            BEQ     20$                     ;Y,N-20$
1423                    001                     .IF DF MM
1424                                            .IFF
1425  002666' 011501                            MOV     (R5),R1                 ;GET PROGRAM TBL ADR
1426                                            .IFT
1427                                            MOV     R0,-(SP)                ;SAVE OPSW VALUE
1428                                            MOV     (R5),CWA                ;STORE PROG'S HI 16 BIT ADR
1429                                            JSR     R5,G22BAD               ;GENERATE 22 BIT ADR
1430                                            .WORD   CWA-.
1431                                            JSR     PC,CVTVAD               ;CONVERT IT TO A VIRTUAL ADR
1432                                            MOV     (SP)+,R0                ;RESTORE OPSW VALUE
1433                    000                     .ENDC
1434  002670' 010061  000002                    MOV     R0,POPSW(R1)            ;MOVE NEW VALUE TO OPSW
1435  002674' 050504                            BIS     R5,R4                   ;SET PROG FND FLAG
1436  002676' 005725                    20$:    TST     (R5)+                   ;INCR TO NEXT TBL ENTRY
1437  002700' 022715  177777                    CMP     #177777,(R5)            ;END OF THE TABLE?
1438  002704' 001366                            BNE     10$                     ;Y,N-10$
1439  002706' 005704                            TST     R4                      ;WERE PROG'S FOUND?
1440  002710' 001402                            BEQ     COPER                   ;Y,N-COPER
1441  002712' 000167  177044            CCFPBJ: JMP     CCFMTB                  ;GO CHECK FOR NO MORE DATA
1442  002716' 004567  003302            COPER:  JSR     R5,CTYPE                ;ISSUE 'NO PROG'S FND' MSG
1443  002722' 000002                            .WORD   002
```

```
1444  002724' 004041                          .WORD   CERM14-.
1445
1446                                          ; ** OPSWN FORMAT **
1447
1448  002726' 010267  176530          COPSWN: MOV     R2,CPNTR                ;SET UP START ADR OF #
1449  002732' 004567  005546                  JSR     R5,CVPNUM               ;GO VALIDATE THIS FROG #
1450                    001                   .IF DF MM
1451                                          JSR     PC,UPAPAR               ;MAP PAR'S TO USER'S PROG AREA
1452                    000                   .ENDC
1453  002736' 005703                          TST     R3                      ;PROG EXIST FOR THIS #?
1454  002740' 001002                          BNE     30$                     ;N,Y-30$
1455  002742' 000167  176772                  JMP     CCFPDE                  ;GO REPORT ERROR
1456  002746' 004567  005262          30$:    JSR     R5,CSCNCB               ;SCAN & CONVERT OPSW VALUE
1457  002752' 010063  000002                  MOV     R0,POPSW(R3)            ;MOVE OPSW TO THIS PROG
1458  002756' 000755                          BR      CCFPBJ                  ;EXIT
```

```
                                        .SBTTL  RUN, DISPLAY, & REPORT COMMAND ROUTINES

                                ;       ******************
                                ;       *                *
                                ;       *  RUN COMMAND   *
                                ;       *                *
                                ;       ******************

002760' 032713  000100          CRUN:   BIT     #OCPRES,(R3)            ;OBJ CODE PRESENT FOR THIS PROG?
002764' 001505                          BEQ     CRUER1                  ;Y,N-CRUER1
002766' 005767  031222                  TST     CSYSFW                  ;CONS REQ BEING PROCESSED?
002772' 100015                          BPL     5$                      ;Y,N-5$
002774' 005713                          TST     (R3)                    ;THIS PROG ALREADY ACTIVE?
002776' 100013                          BPL     5$                      ;Y,N-5$
003000' 021467  031662                  CMP     (R4),CPGCPT             ;THIS PROG IN CONTROL?
003004' 001511                          BEQ     CRUER4                  ;N,Y-CRUER4
003006' 032713  000040                  BIT     #SETDED,(R3)            ;DID IT SET "PRT DEV DEDICATED"?
003012' 001405                          BEQ     5$                      ;Y,N-5$
003014' 042767  060000  031172          BIC     #CTUDED+CRUDED,CSYSFW   ;RESET SYSTEM'S DEDICATED FLAGS
003022' 042713  000040                  BIC     #SETDED,(R3)            ;RESET PROG'S DEDICATED FLAG
003026' 105063  000034          5$:     CLRB    PDPNTR(R3)              ;CLEAR DEV # POINTER
003032' 004767  032332                  JSR     PC,CPGCDN               ;POINT AT FIRST DEV #
003036' 111263  000035                  MOVB    (R2),PCURDV(R3)         ;STORE IT AS CURR DEV
003042' 004567  032336                  JSR     R5,CPGHSK               ;GO DO PROG HOUSEKEEPING
003046' 000001                          .WORD   1                       ;  UNCOND. DO HSKP FLAG
003050' 042713  000036                  BIC     #URSTOP+ERSTOP+WT4IOT+CTPRIO,(R3)  ;RESET STOP & WAIT FLAGS
003054' 052713  100000                  BIS     #ACTIVE,(R3)            ;SET IT'S ACTIVE FLAG
                   001                  .IF DF MM
                                        JSR     PC,MAPUPG               ;MAP USER'S PAR'S & PDR'S STORAGE
                   000                  .ENDC
003060' 005767  005130                  TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
003064' 100002                          BPL     15$                     ;N,Y-15$
003066' 000167  175376          10$:    JMP     CGORUN                  ;GO INITIATE PROG EXECUTION
003072' 122777  000101  176362  15$:    CMPB    #'A,@CPNTR              ;ITS FIRST CHAR AN 'A'?
003100' 001120                          BNE     COMRET                  ;Y,N-COMRET
003102' 004567  005114                  JSR     R5,CSCN                 ;SCAN OVER THE 'AT'
003106' 042713  100000                  BIC     #ACTIVE,(R3)            ;RESET THE ACTIVE FLAG FOR NOW
003112' 005702                          TST     R2                      ;IS THERE A LINE #?
003114' 100435                          BMI     CRUER2                  ;Y,N-CRUER2
003116' 004567  005120                  JSR     R5,CSCNCD               ;SCAN & CONVERT IT TO PACKED DEC
003122' 105720                          TSTB    (R0)+                   ;LINE # TOO BIG?
003124' 001035                          BNE     CRUER3                  ;N,Y-CRUER3
003126' 016301  000022                  MOV     PSRCST(R3),R1           ;GET SOURCE STMNTS START ADR
                   001                  .IF DF MM
                                        BIS     #P4CONS,R1              ;ADJ TO KERNEL'S PAR'S
                   000                  .ENDC
003132' 111102                  20$:    MOVB    (R1),R2                 ;GET STMNT LENGTH
003134' 026110  000002                  CMP     2(R1),(R0)              ;LINE #'S MATCH?
003140' 001406                          BEQ     30$                     ;N,Y-30$
003142' 122761  000240  000001          CMPB    #240,1(R1)              ;THIS THE 'END' STMNT?
003150' 001423                          BEQ     CRUER3                  ;N,Y-CRUER3
003152' 060201                          ADD     R2,R1                   ;INCR TO NXT STMNT
003154' 000766                          BR      20$                     ;GO PROCESS NXT STMNT
003156' 016163  000004  000236  30$:    MOV     4(R1),PUSRPC(R3)        ;STORE THIS STMNT'S START ADR
003164' 052713  100000                  BIS     #ACTIVE,(R3)            ;SET PROG'S ACTIVE FLAG
```

```
1516  003170' 005767 005020                TST     CSCFWD          ;ANOTHER OPERAND IN CMND?
1517  003174' 100734                       BMI     10$             ;Y,N-10$ -- GO TO EXECUTION
1518  003176' 000461                       BR      COMRET          ;GO CK FOR ANOTHER PROG #
1519
1520
1521  003200' 004567 003020 CRUER1: JSR     R5,CTYPE        ;GO ISSUE NO OBJ CODE MSG
1522  003204' 000006                .WORD   006
1523  003206' 003575                .WORD   CERM15-.
1524  003210' 004567 003010 CRUER2: JSR     R5,CTYPE        ;REPORT NO LINE # ERROR
1525  003214' 000046                .WORD   046
1526  003216' 003601                .WORD   CERM16-.
1527  003220' 004567 003000 CRUER3: JSR     R5,CTYPE        ;REPORT INV LINE # ERROR
1528  003224' 000116                .WORD   116
1529  003226' 003571                .WORD   CERM16-.
1530  003230' 004567 002770 CRUER4: JSR     R5,CTYPE        ;REPORT PROG IN CONTROL
1531  003234' 000006                .WORD   006
1532  003236' 003736                .WORD   CERM30-.
1533
1534
1535                          ;       **********************
1536                          ;       *                    *
1537                          ;       *  DISPLAY COMMAND   *
1538                          ;       *                    *
1539                          ;       **********************
1540
1541
1542  003240' 016367 000006 004312 CDISPL: MOV     PASCIN(R3),CDISPN       ;MOVE PROG # TO MSG
1543  003246' 010300                       MOV     R3,R0                   ;GET ADR OF PROG'S NAME
1544  003250' 062700 000010                ADD     #PNAME,R0
1545  003254' 010701                       MOV     PC,R1                   ;GET ADR OF PLACE IN MSG
1546  003256' 062701 004306                ADD     #CDNAME-.,R1
1547  003262' 012021                       MOV     (R0)+,(R1)+             ;MOVE NAME TO MSG
1548  003264' 012021                       MOV     (R0)+,(R1)+
1549  003266' 011011                       MOV     (R0),(R1)
1550  003270' 004567 032304                JSR     R5,CLIST                ;ISSUE DISPLAY HEADER MSG
1551  003274' 004253                       .WORD   CDSHDR-.
1552  003276' 000040                       .WORD   32.
1553  003300' 005767 004710                TST     CSCFWD          ;ANOTHER OPERAND IN CMND?
1554  003304' 100414                       BMI     50$             ;Y,N-50$
1555  003306' 122777 000103 176146         CMPB    #'C,@CPNTR      ;IT'S FIRST CHAR A "C"?
1556  003314' 001010                       BNE     50$             ;Y,N-50$
1557  003316' 004567 004700                JSR     R5,CSCN         ;SCAN OVER THE WORD "CODE"
1558  003322' 032713 000100                BIT     #OCPRES,(R3)    ;OBJ CODE PRESENT?
1559  003326' 001724                       BEQ     CRUER1          ;Y,N-CRUER1
1560  003330' 004567 021142                JSR     R5,DISPCD       ;GO LIST SOURCE & OBJ CODE
1561  003334' 000402                       BR      COMRET          ;RETURN FOR NXT PROG # CHECK
1562  003336' 004567 021126        50$:    JSR     R5,DISP         ;GO LIST ONLY SOURCE STMNTS
1563  003342' 000167 176402        COMRET: JMP     CCFMTA          ;GO CK FOR ANOTHER PROG #
```

```
1565                                    ;       ********************
1566                                    ;       *                  *
1567                                    ;       *  REPORT COMMAND  *
1568                                    ;       *                  *
1569                                    ;       ********************
1570
1571
1572  003346' 016746  032150            CREPOR: MOV     CRPFLG,-(SP)            ;SAVE FLAG IN CASE ITS A USER INT
1573  003352' 012767  100003  032142            MOV     #100003,CRPFLG          ;SET FLG TO DO BOTH REPORTS
1574  003360' 005767  004630                    TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
1575  003364' 100420                            BMI     30$                     ;Y,N-30$
1576  003366' 122777  000123  176066            CMPB    #'S,@CPNTR              ;IS ITS FIRST CHAR AN 'S'?
1577  003374' 001003                            BNE     20$                     ;Y,N-20$
1578  003376' 005367  032120                    DEC     CRPFLG                  ;LEAVE STATUS REPORT FLAG SET
1579  003402' 000407                            BR      25$                     ;GO TO DO REPORT
1580  003404' 122777  000103  176050  20$:      CMPB    #'C,@CPNTR              ;IS IT A 'C'?
1581  003412' 001005                            BNE     30$                     ;Y,N-30$
1582  003414' 012767  100001  032100            MOV     #100001,CRPFLG          ;SET COUNTS REPORT FLAG
1583  003422' 004567  004574            25$:    JSR     R5,CSCN                 ;SCAN OVER VALID OPERAND
1584  003426' 005763  000300            30$:    TST     PTLGTH+DERPAD(R3)       ;IS THERE AN ADR FOR ERROR REPORT?
1585  003432' 001005                            BNE     40$                     ;N,Y-40$
1586  003434' 004567  002564                    JSR     R5,CTYPE                ;ISSUE NO REPT AVAIL MSG
1587  003440' 000604                            .WORD   604
1588  003442' 004145                            .WORD   CRENPA-.
1589  003444' 000402                            BR      50$                     ;GO TO CMND EXIT
1590              001                           .IF DF MM
1591                                            .IFF
1592  003446' 004767  032032            40$:    JSR     PC,CGORPT               ;GO HAVE REPORT ISSUED
1593                                            .IFT
1594                                    40$:    MOV     SPPAR6,@#KPAR6          ;POINT PAR 6 TO SHARED CODE
1595                                            JSR     PC,CGORPT               ;GO HAVE REPORT ISSUED
1596              000                           .ENDC
1597  003452' 012667  032044            50$:    MOV     (SP)+,CRPFLG            ;RESTORE ORG FLAG
1598  003456' 000731                            BR      COMRET                  ;GO CK FOR NXT PROG #
```

```
                                        .SBTTL  FILL COMMAND ROUTINE

                                ;       ******************
                                ;       *                *
                                ;       *  FILL COMMAND  *
                                ;       *                *
                                ;       ******************

003460' 122777  000103  176110  CFILL:  CMPB    #'C,@CDSTAD             ;OPERAND = COM?
003466' 001007                          BNE     2$                      ;Y,N-2$
003470' 010704                          MOV     PC,R4                   ;SET UP COM00 AS START ADR
003472' 062704  036346                  ADD     #LCOM0-.,R4
003476' 010405                          MOV     R4,R5                   ;SET UP COM09+2 AS END ADR
003500' 062705  000024                  ADD     #20.,R5
003504' 000412                          BR      4$                      ;GO CK FOR #
003506' 122777  000102  176062  2$:     CMPB    #'B,@CDSTAD             ;OPERAND = BUF?
003514' 001156                          BNE     CFIER1                  ;Y,N-CFIER1
003516' 010704                          MOV     PC,R4                   ;SET START ADR TO BUF 0 BEGIN &
003520' 062704  035720                  ADD     #CBUF00-.,R4            ;END ADR TO END OF BUF 15
003524' 010405                          MOV     R4,R5
003526' 062705  000400                  ADD     #256.,R5
003532' 004567  007734          4$:     JSR     R5,CISDNM               ;GO ISOLATE BUFFER/COM #
003536' 005701                          TST     R1                      ;WAS THERE A #?
003540' 001435                          BEQ     15$                     ;Y,N-15$
003542' 004567  036602                  JSR     R5,ASDECO               ;CONVERT IT TO BINARY
003546' 000314                          .WORD   CFIER2-.
003550' 004567  036342                  JSR     R5,DECBIN
003554' 000306                          .WORD   CFIER2-.
003556' 122777  000103  176012          CMPB    #'C,@CDSTAD             ;IS THIS "COM" FORMAT?
003564' 001010                          BNE     6$                      ;Y,N-6$
003566' 020027  000011                  CMP     R0,#9.                  ;COM # TOO BIG?
003572' 101133                          BHI     CFIER2                  ;N,Y-CFIER2
003574' 006300                          ASL     R0                      ;MULT # BY 2
003576' 060004                          ADD     R0,R4                   ;SET UP COM START ADR
003600' 010405                          MOV     R4,R5                   ;SET UP COM END ADR
003602' 005725                          TST     (R5)+
003604' 000413                          BR      15$                     ;GO PROCESS DATA
003606' 020027  000017          6$:     CMP     R0,#15.                 ;BUF # TOO LARGE?
003612' 101123                          BHI     CFIER2                  ;N,Y-CFIER2
003614' 006300                          ASL     R0                      ;COMPUTE BUFFER'S START & END
003616' 006300                          ASL     R0                      ;ADR'S
003620' 006300                          ASL     R0
003622' 006300                          ASL     R0
003624' 060004                          ADD     R0,R4
003626' 010405                          MOV     R4,R5
003630' 062705  000020                  ADD     #16.,R5
003634' 004567  004362          15$:    JSR     R5,CSCN                 ;SCAN NXT CMND OPERAND
003640' 122777  000127  175730          CMPB    #'W,@CDSTAD             ;IS IT THE 'WITH' OPERAND?
003646' 001111                          BNE     CFIER3                  ;Y,N-CFIER3
003650' 005702                          TST     R2                      ;ANOTHER OPER PRESENT?
003652' 100513                          BMI     CFIER4                  ;Y,N-CFIER4
003654' 005003                          CLR     R3                      ;CLEAR CHAR CNT
003656' 005002                          CLR     R2                      ;RESET INCLUDE CR/LF FLAG
003660' 122777  000052  175574          CMPB    #'*,@CPNTR              ;FIRST CHAR OF NXT WD = *?
003666' 001435                          BEQ     55$                     ;N,Y-55$
```

```
1656  003670' 122777 000043 175564        CMPB    #'#,@CPNTR              ;IS IT A NBR SIGN?
1657  003676' 001433                      BEQ     60$                     ;N,Y-60$
1658  003700' 122777 000120 175554        CMPB    #'P,@CPNTR              ;IS IT A "P" FOR PAT?
1659  003706' 001505                      BEQ     CFIPAT                  ;N,Y-CFIPAT
1660
1661                                      ;OCTAL/BINARY FORMAT
1662
1663  003710' 004567 004320        20$:   JSR     R5,CSCNCB               ;SCAN & CONVERT DATA WORD
1664  003714' 010024                      MOV     R0,(R4)+                ;MOV DATA TO THE BUFFER
1665  003716' 005723                      TST     (R3)+                   ;ADD 2 TO CHAR CNT
1666  003720' 020405                      CMP     R4,R5                   ;IS THE BUFFER FULL?
1667  003722' 001552                      BEQ     CFIEX                   ;N,Y-CFIEX
1668  003724' 005767 004264               TST     CSCFWD                  ;ANOTHER DATA WORD PRESENT?
1669  003730' 100367                      BPL     20$                     ;N,Y-20$
1670  003732' 005703                30$:  TST     R3                      ;CHAR CNT=0?
1671  003734' 001462                      BEQ     CFIER4                  ;N,Y-CFIER4
1672  003736' 010401                      MOV     R4,R1                   ;SET UP BUF START ADR
1673  003740' 160301                      SUB     R3,R1
1674  003742' 010302                40$:  MOV     R3,R2                   ;SET UP WORK CHAR CNT
1675  003744' 010100                      MOV     R1,R0                   ;SET UP WORK BUFFER START ADR
1676  003746' 112024                50$:  MOVB    (R0)+,(R4)+             ;PROPAGATE A CHARACTER
1677  003750' 020405                      CMP     R4,R5                   ;BUFFER FULL?
1678  003752' 001536                      BEQ     CFIEX                   ;N,Y-CFIEX
1679  003754' 005302                      DEC     R2                      ;DECR CHAR CNT BY 1
1680  003756' 001373                      BNE     50$                     ;CNT=0? (Y,N-50$)
1681  003760' 000770                      BR      40$                     ;GO START AT BEGINNING OF DATA
1682
1683                                      ;ASCII FORMAT
1684
1685  003762' 005202                55$:  INC     R2                      ;SET 'INCLUDE CR/LF' FLAG
1686  003764' 005745                      TST     -(R5)                   ;ALLOW ROOM FOR CR/LF
1687  003766' 016700 175470         60$:  MOV     CPNTR,R0                ;GET DATA ADR
1688  003772' 005200                      INC     R0                      ;POINT PASS THE "*/#"
1689  003774' 020405                70$:  CMP     R4,R5                   ;BUFFER FULL?
1690  003776' 001411                      BEQ     80$                     ;N,Y-80$
1691  004000' 122710 000015               CMPB    #015,(R0)               ;CHAR A CARR RET?
1692  004004' 001406                      BEQ     80$                     ;N,Y-80$
1693  004006' 122710 000012               CMPB    #012,(R0)               ;IS IT A LINE FEED?
1694  004012' 001403                      BEQ     80$                     ;N,Y-80$
1695  004014' 112024                      MOVB    (R0)+,(R4)+             ;MOVE CHAR TO THE BUFFER
1696  004016' 005203                      INC     R3                      ;ADD 1 TO CHAR CNT
1697  004020' 000765                      BR      70$                     ;RETURN FOR NXT CHAR
1698  004022' 005702                80$:  TST     R2                      ;INCLUDE CR/LF?
1699  004024' 001407                      BEQ     90$                     ;Y,N-90$
1700  004026' 112724 000015               MOVB    #015,(R4)+              ;SET CR/LF IN OUTPUT AREA
1701  004032' 112724 000012               MOVB    #012,(R4)+
1702  004036' 062703 000002               ADD     #2,R3                   ;INCLUDE THEM IN CHAR CNT
1703  004042' 005725                      TST     (R5)+                   ;RE-ADJUST END ADR
1704  004044' 020405                90$:  CMP     R4,R5                   ;BUFFER FULL?
1705  004046' 001331                      BNE     30$                     ;Y,N-30$
1706  004050' 000477                      BR      CFIEX                   ;GO TO CMND EXIT
1707
1708
1709  004052' 004567 002146         CFIER1: JSR     R5,CTYPE                ;REPORT NO BUF OPERAND
1710  004056' 000142                        .WORD   142
1711  004060' 002647                        .WORD   CERM10-.
```

```
1712  004062' 004567 002136          CFIER2: JSR     R5,CTYPE                ;REPORT INV BUF #
1713  004066' 000112                          .WORD   112
1714  004070' 002647                          .WORD   CERM11-.
1715  004072' 004567 002126          CFIER3: JSR     R5,CTYPE                ;REPORT "WITH" NOT SPEC
1716  004076' 000142                          .WORD   142
1717  004100' 002651                          .WORD   CERM12-.
1718  004102' 004567 002116          CFIER4: JSR     R5,CTYPE                ;REPORT NO DATA SPEC
1719  004106' 000042                          .WORD   042
1720  004110' 002650                          .WORD   CERM13-.
1721  004112' 004567 002106          CFIER5: JSR     R5,CTYPE                ;REPORT INV PAT #
1722  004116' 000112                          .WORD   112
1723  004120' 003115                          .WORD   CERM32-.
1724
1725
1726                                          ;PATTERN FORMAT
1727
1728  004122' 010467 000114          CFIPAT: MOV     R4,110$                 ;STORE DATA ADR
1729  004126' 160405                  SUB     R4,R5                   ;COMPUTE BYTE COUNT
1730  004130' 010567 000110                  MOV     R5,112$                 ;STORE BYTE COUNT
1731  004134' 004567 004062                  JSR     R5,CSCN                 ;SCAN THE WORD "PAT"
1732  004140' 004567 007326                  JSR     R5,CISDNM               ;SCAN WORD FOR DECIMAL #
1733  004144' 005701                  TST     R1                      ;WAS THERE A #?
1734  004146' 001014                  BNE     100$                    ;N,Y-100$
1735  004150' 122710 000101                  CMPB    #'A,(R0)                ;IS THIS "PATA"?
1736  004154' 001003                  BNE     95$                     ;Y,N-95$
1737  004156' 012700 000012                  MOV     #10.,R0                 ;SET UP PATA VALUE
1738  004162' 000417                  BR      105$                    ;GO DO PATA
1739  004164' 122710 000102          95$:    CMPB    #'B,(R0)                ;IS IT "PATB"?
1740  004170' 001350                  BNE     CFIER5                  ;Y,N-CFIER5
1741  004172' 012700 000013                  MOV     #11.,R0                 ;SET UP PATB VALUE
1742  004176' 000411                  BR      105$                    ;GO DO PATB
1743  004200' 004567 036144          100$:   JSR     R5,ASDECO               ;CONVERT # TO BINARY
1744  004204' 177706                          .WORD   CFIER5-.
1745  004206' 004567 035704                  JSR     R5,DECBIN
1746  004212' 177700                          .WORD   CFIER5-.
1747  004214' 020027 000014                  CMP     R0,#LPATEN-LPATOP/2     ;PATTERN # TOO LARGE?
1748  004220' 103334                  BHIS    CFIER5                  ;N,Y-CFIER5
1749  004222' 006300                  105$:   ASL     R0                      ;MULT # BY 2 TO GET DISPL.
1750  004224' 060700                  ADD     PC,R0                   ;INDEX INTO TABLE
1751  004226' 062700 035636                  ADD     #LPATOP-.,R0
1752  004232' 010067 000010                  MOV     R0,114$                 ;STORE ADR OF PATTERN'S REL ADR
1753  004236' 004567 032064                  JSR     R5,LFILP                ;GO GENERATE THE PATTERN
1754  004242' 000000                  110$:   .WORD   XXXX                    ;  ADDRESS
1755  004244' 000000                  112$:   .WORD   XXXX                    ;  BYTE COUNT
1756  004246' 000000                  114$:   .WORD   XXXX                    ;  REL PATTERN ADR
1757  004250' 000167 174112          CFIEX:  JMP     CNCMND                  ;RETURN FOR NEXT CMND
```

```
                                        .SBTTL  MEMORY MAP COMMAND ROUTINE

                                ;       **********************
                                ;       *                    *
                                ;       *  MEM MAP COMMAND   *
                                ;       *                    *
                                ;       **********************

004254' 004567  001744          CMAP:   JSR     R5,CTYPE                ;ISSUE MEM MAP HEADER MSG
004260' 000600                          .WORD   600
004262' 003134                          .WORD   CMMHDR-.
            001                         .IF DF MM
                                        TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
                                        BMI     CMPALL                  ;Y,N-CMPALL
                                        ADD     #CMP1PG-CMAP,CR2ADR     ;ADJ RETURN ADR OF CCFMTA S/R
                                        JMP     CCFMT1                  ;GO DO COM PROC FOR PROG #
                                CMP1PG: MOV     R4,-(SP)                ;SET 1 PROG MAP FLAG
                                        BR      CMPPPG                  ;GO PROCESS THIS PROG
                                CMPALL: CLR     -(SP)                   ;RESET 1 PROG MAP FLG (WILL DO ALL)
            000                         .ENDC
004264' 010704                          MOV     PC,R4                   ;SET UP CTRL ROUT TBL ADR
004266' 062704  027730                  ADD     #CRTBL-.,R4
004272' 005714                  CMPCKE: TST     (R4)                    ;ENTRY IN USE?
004274' 001005                          BNE     CMPPPG                  ;N,Y-CMPPPG
004276' 005724                  CMPNXP: TST     (R4)+                   ;INCR TO NXT ENTRY
004300' 022714  177777                  CMP     #177777,(R4)            ;END OF TBL?
004304' 001372                          BNE     CMPCKE                  ;Y,N-CMPCKE
004306' 000436                          BR      CMPPFM                  ;GO DO FREE MEM MSGS
004310' 011400                  CMPPPG: MOV     (R4),R0                 ;GET PROG START ADR
004312' 004567  034604                  JSR     R5,BINASC               ;CONVERT IT TO ASCII & PUT
004316' 003136                          .WORD   CMMSTA-.                ;IN MSG
            001                         .IF DF MM
                                        .IFF
004320' 011400                          MOV     (R4),R0                 ;GET PROG END ADR
004322' 066000  000026                  ADD     PLNGTH(R0),R0
004326' 005300                          DEC     R0
                                        .IFT
                                        MOV     (R4),R3                 ;SET HI 16 BITS IN CORRECT REG
                                        JSR     PC,UPAPAR               ;MAP KERNEL PAR'S TO THIS PROG
                                        MOVB    #40,CMMFPI              ;RESET FIXED PROG CHAR IN MSG
                                        MOV     CRTEXT(R4),R0           ;GET PROG'S LENGTH IN 32 WD BLKS
                                        BPL     10$                     ;FIXED ADR PROG? (Y,N-10$)
                                        MOVB    #'F,CMMFPI              ;SET AN "F" IN THE MSG
                                        BIC     #FIXADR,R0              ;RESET FIXED ADR FLG
                                10$:    DEC     R0                      ;ADJ IT
                                        ADD     (R4),R0                 ;ADD IN START ADR
                                        .IFTF
004330' 004567  034566                  JSR     R5,BINASC               ;CONVERT TO ASCII & PLACE
004334' 003132                          .WORD   CMMEND-.                ;IN MSG
                                        .IFF
004336' 011400                          MOV     (R4),R0                 ;GET PROG TBL ADR AGAIN
                                        .IFT
                                        MOV     R3,R0                   ;GET PROG TBL ADR AGAIN
            000                         .ENDC
004340' 062700  000010                  ADD     #PNAME,R0               ;POINT AT PROG'S NAME
```

```
1815  004344' 010701                  MOV     PC,R1                   ;GET ADR OF PLACE IN MSG
1816  004346' 062701  003130          ADD     #CMMPNM-.,R1
1817  004352' 012021                  MOV     (R0)+,(R1)+             ;MOVE PROG'S NAME TO MSG
1818  004354' 012021                  MOV     (R0)+,(R1)+
1819  004356' 012011                  MOV     (R0)+,(R1)
1820  004360' 016001  000014          MOV     PMDLCD-PNAME-6(R0),R1   ;GET PROG'S DEV MDL CODE
1821  004364' 004567  007156          JSR     R5,FDVNAM               ;GET DEV NAME FOR THIS MDL CODE &
1822  004370' 003116                  .WORD   CMMMDL-.                ;PUT IN THE MSG
1823  004372' 004567  001626          JSR     R5,CTYPE                ;GO ISSUE PROG # & START & END
1824  004376' 000404                  .WORD   404                     ;ADR'S & NAME & MDL MSG
1825  004400' 003051                  .WORD   CMMMSG-.
1826              001                 .IF DF MM
1827                                  .IFF
1828  004402' 000735                  BR      CMPNXP                  ;GO PROCESS NXT PROG
1829                                  .IFT
1830                                  TST     (SP)                    ;DOING ONLY 1 PROG?
1831                                  BEQ     CMPNXP                  ;Y,N-CMPNXP
1832                                  ADD     #PUPARS,R3              ;POINT AT PAR STORAGE IN PROG TBL
1833                                  MOV     R3,R5                   ;SET UP ADR OF PDR REGS
1834                                  ADD     #PUPDRS-PUPARS,R5
1835                                  TST     CSYSFW                  ;CONS REG BEING PROCESSED?
1836                                  BPL     20$                     ;Y,N-20$
1837                                  TST     CPGCPT                  ;USER PROG IN CONTROL?
1838                                  BEQ     20$                     ;Y,N-20$
1839                                  CMP     R4,CPGCPE               ;IS IT THIS PROG?
1840                                  BNE     20$                     ;Y,N-20$
1841                                  MOV     #UPAR0,R3               ;SET UP USER PAR REG ADR
1842                                  MOV     #UPDR0,R5               ;SET UP USER PDR REG ADR
1843                          20$:    JSR     PC,PRTPAD               ;GO PRINT PAR & PDR VALUES
1844                                  MOV     #P4CONS+PRDIOX,R3       ;POINT AT 18/22 BIT ABS ADRS IN PROG TBL
1845                                  JSR     PC,PRTAAD               ;DISPLAY RDIO/WRIO ABS ADRS
1846                                  BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
1847                                  BEQ     25$                     ;Y,N-25$
1848                                  MOV     #P4CONS,R3              ;RESTORE PROG TBL PNTR
1849                                  TST     PUBMAP(R3)              ;THIS PROG USING UNIBUS MAP?
1850                                  BEQ     25$                     ;Y,N-25$
1851                                  ADD     #PRDIOV,R3              ;POINT AT 18 BIT VIRT ADRS
1852                                  JSR     PC,PRTVAD               ;DISPLAY RDIO/WRIO VIRT ADRS
1853                                  MOV     #P4CONS+PUBMAP,R3       ;SET UP NEW TBL PNTR
1854                                  MOVB    (R3)+,R0                ;GET 1ST REG #
1855                                  MOVB    (R3),R1                 ;GET # OF REG BITS
1856                                  CMP     R1,#3                   ;CK HOW MANY REGS
1857                                  BLO     24$                     ;BR IF 1
1858                                  BEQ     23$                     ;BR IF 2
1859                                  SUB     #3,R1                   ;ADJ FOR 3 REGS
1860                          23$:    DEC     R1                      ;ADJ FOR 2 REGS
1861                          24$:    JSR     PC,PRTUMR               ;DISPLAY PROG'S UNIBUS MAP REGS
1862                          25$:    JSR     R5,CTYPE                ;ISSUE A PAIR OF CR/LF'S
1863                                  .WORD   600
1864                                  .WORD   MMCRLF-.
1865                          30$:    TST     CSCFWD                  ;IS THERE ANOTHER PROG #?
1866                                  BMI     CMPPFM                  ;Y,N-CMPPFM
1867                                  TST     (SP)+                   ;REMOVE 1 PROG FLAG
1868                                  JMP     CCFMT1                  ;GO VALIDATE ITS #
1869                                  .IFF
1870  004404' 016700  175036  CMPPFM: MOV     CFMST,R0                ;GET FREE MEM START ADR
```

```
1871  004410' 010003                    MOV     R0,R3                   ;SAVE IT
1872  004412' 004567 034504             JSR     R5,BINASC               ;CONVERT TO ASCII & PUT
1873  004416' 003036                    .WORD   CMMSTA-.                ;IN MSG
1874  004420' 016700 175024             MOV     CFMEND,R0               ;GET END OF FREE MEM ADR
1875  004424' 005200                    INC     R0
1876  004426' 010004                    MOV     R0,R4                   ;SAVE IT
1877  004430' 004567 034466             JSR     R5,BINASC               ;CONVERT TO ASCII & PUT
1878  004434' 003032                    .WORD   CMMEND-.                ;IN MSG
1879  004436' 004567 001562             JSR     R5,CTYPE                ;ISSUE FREE MEMORY MSG
1880  004442' 000420                    .WORD   420
1881  004444' 002771                    .WORD   CFMMSG-.
1882  004446' 000040                    .WORD   32.
1883  004450' 005204                    INC     R4                      ;ADD 1 TO END OF MEM ADR
1884  004452' 160304                    SUB     R3,R4                   ;GET SIZE OF "FREE"
1885  004454' 000241                    CLC                             ;DIVIDE IT BY 2
1886  004456' 006004                    ROR     R4
1887  004460' 042704 000001             BIC     #1,R4                   ;MAKE IT EVEN
1888  004464' 060403                    ADD     R4,R3                   ;COMPUTE ADR OF "MIDL"
1889  004466' 010300                    MOV     R3,R0
1890  004470' 004567 034426             JSR     R5,BINASC               ;CONVERT IT TO ASCII &
1891  004474' 003027                    .WORD   CMIDL-.                 ;PUT IN THE MSG
1892  004476' 004567 001522             JSR     R5,CTYPE                ;ISSUE "MIDL" ADR MSG
1893  004502' 000400                    .WORD   400
1894  004504' 003007                    .WORD   CMIDLM-.
1895  004506' 000167 173654             JMP     CNCMND                  ;RETURN FOR NEXT CMND
1896                                    .IFT
1897                            CMPPFM: MOV     PC,R3                   ;SET UP ADR OF MIDL'S 18/22
1898                                    ADD     #MIDLAA-.,R3            ;ABS ADRS
1899                                    JSR     PC,PRTFAA               ;DISPLAY FREE/MIDL ABS ADRS
1900                                    TST     (SP)+                   ;DOING ONLY 1 PROG?
1901                                    BEQ     CMPPRM                  ;Y,N-CMPPRM
1902                                    BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
1903                                    BEQ     CMPPRM                  ;Y,N-CMPPRM
1904                                    MOV     PC,R3                   ;SET UP ADR OF MIDL'S 18
1905                                    ADD     #MIDLVA-.,R3            ;BIT VIRT ADR
1906                                    JSR     PC,PRTFVA               ;DISPLAY FREE/MIDL VIRT ADRS
1907                                    MOV     CFMEND,R2               ;GET FREE MEM END ADR HI 16 BITS
1908                                    INC     R2                      ;ADJ IT
1909                                    SUB     CFMST,R2                ;GET SIZE OF FREE
1910                                    CLR     R1                      ;INITIALIZE # OF REGS
1911                            40$:    INC     R1                      ;ADD 1 TO REG CNT
1912                                    SUB     #200,R2                 ;SUB 4K WORDS
1913                                    BGT     40$                     ;FND FREE'S END? (Y,N-40$)
1914                                    MOV     #3,R0                   ;SET UP FREE'S 1ST REG #
1915                                    JSR     PC,PRTUMR               ;DISPLAY FREE'S UNIMAP REGS
1916                            CMPPRM: MOV     CRMEND,R0               ;GET END OF REAL MEM ADR
1917                                    JSR     R5,BINASC               ;CONVERT TO ASCII &
1918                                    .WORD   CMMEMA-.                ;PUT IN THE MSG
1919                                    JSR     R5,CTYPE                ;DISPLAY REAL MEM END ADR
1920                                    .WORD   400
1921                                    .WORD   CMMERM-.
1922                                    JMP     CNCMND                  ;RETURN FOR NEXT CMND
1923
1924
1925                                    ;PRINT PAR & PDR VALUES
1926
```

```
                                        ;JSR    PC,PRTPAD       S/R CALL
                                        ;
                                        ;R3 = UPARS' ADR
                                        ;R5 = UPDRS' ADR
                                        ;
                                        ;DESTROYS R0,R1,R2,R3

PRTPAD: MOVB    #'A,CMMPAR+3            ;INITIALIZE MSG TO PAR
10$:    MOV     #8.,-(SP)               ;SET UP REG CNT
        MOV     PC,R1                   ;SET UP MSG ADR
        ADD     #CMMPRD-.,R1
20$:    MOV     (R3)+,R0                ;GET PAR/PDR VALUE
        JSR     R5,BINAS1               ;CONVERT DATA TO ASCII & PUT IN MSG
        INC     R1                      ;POINT TO NXT FIELD IN MSG
        DEC     (SP)                    ;DONE ALL REGS?
        BNE     20$                     ;Y,N-20$
        TST     (SP)+                   ;HSKP STK
        JSR     R5,CTYPE                ;DISPLAY THE PAR/PDR MSG
        .WORD   400
        .WORD   CMMPAR-.
        CMPB    #'D,CMMPAR+3            ;JUST DO THE PDR MSG?
        BEQ     30$                     ;N,Y-30$
        MOVB    #'D,CMMPAR+3            ;SET MSG TO PDR
        MOV     R5,R3                   ;SET UP ADR OF PDR REGS
        BR      10$                     ;GO TAILOR & ISSUE PDR MSG
30$:    RTS     PC                      ;EXIT IN-LINE


;DISPLAY RDIO/WRIO 18/22 BIT ABSOLUTE OR VIRTUAL ADDRESSES

                                        ;JSR    PC,PRTAAD       ABS ADRS S/R CALL
                                        ;JSR    PC,PRTVAD       VIRT ADRS S/R CALL
                                        ;
                                        ;R3 = ADR OF 1ST 18/22 BIT ADR
                                        ;
                                        ;DESTROYS R0,R1,R2,R3

PRTAAD: MOV     #"AB,CMMUAD+2           ;TAILOR MSG TO "ABS"
        MOVB    #'S,CMMUAD+4
        BR      PRTAVC                  ;GO TO COMMON POINT

PRTVAD: MOV     #"UB,CMMUAD+2           ;TAILOR MSG TO "UBM"
        MOVB    #'M,CMMUAD+4
PRTAVC: MOV     (R3)+,R2                ;GET RDIO'S 18/22 BIT ADR
        MOV     (R3)+,R0
        JSR     R5,BINAS8               ;CONVERT TO ASCII &
        .WORD   CMMUA1-.                ;PLACE IN MSG
        ADD     #4,R3                   ;POINT AT CORRESPONDING WRIO ADR
        MOV     (R3)+,R2                ;GET WRIO'S 18/22 BIT ADR
        MOV     (R3),R0
        JSR     R5,BINAS8               ;CONVERT TO ASCII &
        .WORD   CMMUA2-.                ;PLACE IN MSG
        JSR     R5,CTYPE                ;ISSUE RDIO & WRIO ADRS MSG
        .WORD   400
        .WORD   CMMUAD-.
        RTS     PC                      ;EXIT IN-LINE
```

```
1983
1984
1985                                    ;DISPLAY FREE/MIDL 18/22 BIT ABSOLUTE OR VIRTUAL ADDRESSES
1986
1987                                            ;JSR    PC,PRTFAA       ABS ADRS S/R CALL
1988                                            ;JSR    PC,PRTFVA       VIRT ADRS S/R CALL
1989                                            ;
1990                                            ;R3 = ADR OF MIDL'S ABS OR VIRT ADR
1991                                            ;
1992                                            ;DESTROYS R0,R1,R2,R3,R4
1993
1994                                    PRTFAA: MOV     #"AB,CFMMSG+2           ;TAILOR MSG TO ABS
1995                                            MOVB    #'S,CFMMSG+4
1996                                            MOV     CFMST,R4                ;GET FREE'S ABS START ADR
1997                                            BR      PRTFAV                  ;GO TO COMMON POINT
1998
1999                                    PRTFVA: MOV     #"UB,CFMMSG+2           ;TAILOR MSG TO "UBM"
2000                                            MOVB    #'M,CFMMSG+4
2001                                            MOV     #600,R4                 ;SET UP FREE'S VIRT START ADR
2002                                    PRTFAV: MOV     R4,R0                   ;SET UP START ADR
2003                                            JSR     R5,BINASC               ;CONVERT TO ASCII & PUT
2004                                            .WORD   CMMSTA-.                ;IN THE MSG
2005                                            MOV     CFMEND,R0               ;GET ABS FREE MEM END ADR
2006                                            SUB     CFMST,R0                ;GET FREE'S SIZE
2007                                            ADD     R4,R0                   ;ADD IN ABS/VIRT START ADR
2008                                            JSR     R5,BINASC               ;CONVERT TO ASCII & PUT
2009                                            .WORD   CMMEND-.                ;IN THE MSG
2010                                            MOV     (R3)+,R2                ;GET MIDL'S ADR
2011                                            MOV     (R3),R0
2012                                            JSR     R5,BINAS8               ;CONVERT TO ASCII & PUT
2013                                            .WORD   CMIDL-.                 ;IN THE MSG
2014                                            JSR     R5,CTYPE                ;ISSUE 1ST PART OF FREE MEM MSG
2015                                            .WORD   420
2016                                            .WORD   CFMMSG-.
2017                                            .WORD   47.
2018                                            JSR     R5,CTYPE                ;ISSUE "MIDL" PORTION OF THE MSG
2019                                            .WORD   400
2020                                            .WORD   CMIDLM-.
2021                                            RTS     PC                      ;EXIT IN-LINE
2022                                            .PAGE
2023
2024
2025                                    ;PRINT THE CONTENTS OF 1 TO 3 UNIBUS MAP REGISTERS
2026
2027                                            ;JSR    PC,PRTUMR       S/R CALL
2028                                            ;
2029                                            ;R0 = 1ST UNIBUS MAP REG #
2030                                            ;R1 = # OF UNIBUS MAP REGS
2031                                            ;
2032                                            ;DESTROYS R0,R1,R2,R3,R4,R5
2033
2034                                    PRTUMR: MOV     R0,R3                   ;SAVE REG #
2035                                            MOV     R0,R4                   ;GET UNIBUS MAP REG ADR
2036                                            ASL     R4                      ;IN R4
2037                                            ASL     R4
2038                                            ADD     #UBMAP0,R4
```

```
2039                                    MOV     PC,R5                   ;POINT AT TBL WITH MSG ADRS
2040                                    ADD     #PRTUMA-.,R5
2041                                    MOV     #3,PRTUBC               ;INITIALIZE MSG LENGTH
2042                                    MOV     R1,-(SP)                ;SET UP LOOP COUNT
2043                            PRTULP: MOV     R3,R0                   ;GET REG # IN CORR REG
2044                                    JSR     PC,CPNAS1               ;CONVERT REG # TO DEC ASCII
2045                                    MOV     PC,R0                   ;GET ADR OF WHERE THE # GOES
2046                            PRTUB1: ADD     (R5)+,R0                ;IN THE MSG
2047                                    MOV     R1,(R0)                 ;STORE IT
2048                                    MOV     (R4)+,R0                ;GET MAP REG CONTENTS
2049                                    MOV     (R4)+,R2
2050                                    MOV     (R5)+,PRTUB2            ;GET ITS MSG ADR
2051                                    JSR     R5,BINAS8               ;CONVERT MAP REG CONTENTS TO
2052                            PRTUB2: .WORD   XXXX                    ;ASCII & PLACE IN THE MSG
2053                                    INC     R3                      ;POINT TO NEXT UB MAP REG
2054                                    ADD     #22.,PRTUBC             ;INCREASE MSG LGTH FOR THIS REG
2055                                    DEC     (SP)                    ;DONE ALL REGS?
2056                                    BNE     PRTULP                  ;Y,N-PRTULP
2057                                    TST     (SP)+                   ;HSKP STK
2058                                    JSR     R5,CTYPE                ;ISSUE UNIBUS MAP CONTENTS MSG
2059                                    .WORD   620
2060                                    .WORD   CMMUBM-.
2061                            PRTUBC: .WORD   XXXX
2062                                    RTS     PC                      ;EXIT IN-LINE
2063
2064                            PRTUMA: .WORD   CMMUB1-PRTUB1
2065                                    .WORD   CMMUB2-PRTUB2
2066                                    .WORD   CMMUB3-PRTUB1
2067                                    .WORD   CMMUB4-PRTUB2
2068                                    .WORD   CMMUB5-PRTUB1
2069                                    .WORD   CMMUB6-PRTUB2
2070                    000             .ENDC
```

```
                                        .SBTTL  /SAVE AND /FETCH COMMAND ROUTINES

                                ;       ********************
                                ;       *                  *
                                ;       *  /SAVE COMMAND   *
                                ;       *                  *
                                ;       ********************

004512' 010300          CSAVE:  MOV     R3,R0                   ;SET UP ADR OF NAME IN
004514' 062700  000010          ADD     #PNAME,R0               ;PROG TBL
004520' 010705                  MOV     PC,R5                   ;SET THE "SAVE" FLAG
004522' 005767  003466  CSVCOM: TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
004526' 100005                  BPL     10$                     ;N,Y-10$
004530' 012702  000006          MOV     #6,R2                   ;SET UP NAME SIZE
004534' 004567  000162          JSR     R5,CMVNAM               ;MOVE NAME TO COMMON AREA
004540' 000405                  BR      20$                     ;GO CK CMND TYPE
004542' 004567  000052  10$:    JSR     R5,CSFCKA               ;CHECK FOR WORD "AS"
004546' 100510                  BMI     CSANNM                  ;IS THERE A NAME? (Y,N-CSANNM)
004550' 004567  000072          JSR     R5,CSACNR               ;SCAN NAME & VALIDATE IT
004554' 005705          20$:    TST     R5                      ;THIS THE ENTER CMND?
004556' 001416                  BEQ     CGOENT                  ;N,Y-CGOENT
004560' 004567  020116          JSR     R5,SAVE                 ;GO TO SAVE ROUTINE
004564' 000167  173576          JMP     CNCMND                  ;RETURN FOR NEXT CMND


                                ;       ********************
                                ;       *                  *
                                ;       *  /FETCH COMMAND  *
                                ;       *                  *
                                ;       ********************

004570' 004567  000056  CFETCH: JSR     R5,CFECNR               ;VALIDATE THE NAME
004574' 004567  000020          JSR     R5,CSFCKA               ;CHECK FOR WORD "AS"
004600' 100503                  BMI     CSANPN                  ;PROG # FOLLOW? (Y,N-CSANPN)
004602' 004567  003676          JSR     R5,CVPNUM               ;VALIDATE PROG #
                        001             .IF DF MM
                                        JSR     PC,UPAPAR       ;MAP PAR'S TO USER'S PROG AREA
                        000             .ENDC
004606' 052767  040000  174630  CGOENT: BIS     #CFTCHF,CRFLGW  ;SET THE FETCH FLAG
004614' 000167  175242          JMP     CENTPA                  ;GO TO 'ENTER' CMND ROUTINE


                                ;       **** SAVE/FETCH COMMON S/R'S ****

004620' 005767  003370  CSFCKA: TST     CSCFWD                  ;ANOTHER WORD IN CMND?
004624' 100455                  BMI     CSANAS                  ;Y,N-CSANAS
004626' 004567  003370          JSR     R5,CSCN                 ;SCAN FOR THE WORD "AS"
004632' 122777  000101  174736  CMPB    #'A,@CDSTAD             ;FIRST CHAR AN "A"?
004640' 001047                  BNE     CSANAS                  ;Y,N-CSANAS
004642' 005702                  TST     R2                      ;CHECK FOR ANOTHER OPERAND
004644' 000205          CMVNEX: RTS     R5                      ;EXIT IN-LINE


004646' 004567  003350  CSACNR: JSR     R5,CSCN                 ;SCAN PROGRAM NAME
```

```
2128  004652' 020127 000006        CFECNR: CMP     R1,#6                   ;IS IT > 6 CHAR'S LONG?
2129  004656' 101050                       BHI     CSAINM                  ;N,Y-CSAINM
2130  004660' 010102                       MOV     R1,R2                   ;SAVE CHAR CNT
2131  004662' 121027 000132        10$:    CMPB    (R0),#'Z                ;IS CHARACTER OF NAME 0-9 OR
2132  004666' 101044                       BHI     CSAINM                  ;A-Z?
2133  004670' 121027 000101                CMPB    (R0),#'A                ;Y,N-CSAINM
2134  004674' 103006                       BHIS    20$
2135  004676' 121027 000071                CMPB    (R0),#'9
2136  004702' 101036                       BHI     CSAINM
2137  004704' 121027 000060                CMPB    (R0),#'0
2138  004710' 103433                       BLO     CSAINM
2139  004712' 005301                20$:    DEC     R1                      ;DECR CHAR CNT
2140  004714' 001402                       BEQ     CMVNAM                  ;CK ALL CHAR'S? (N,Y-CMVNAM)
2141  004716' 005300                       DEC     R0                      ;POINT AT NXT CHAR
2142  004720' 000760                       BR      10$                     ;RETURN FOR NXT CHAR
2143  004722' 012746 000006        CMVNAM: MOV     #6,-(SP)                ;SET UP MAX # OF CHAR'S
2144  004726' 160216                       SUB     R2,(SP)                 ;GET # OF FILL SPACES
2145  004730' 010701                       MOV     PC,R1                   ;SET UP STORAGE ADR
2146  004732' 062701 173046                ADD     #CLOCZ-.,R1
2147  004736' 112021                40$:    MOVB    (R0)+,(R1)+             ;MOVE CHAR TO OUTPUT STORAGE
2148  004740' 005302                       DEC     R2                      ;DECR CHAR CNT
2149  004742' 001375                       BNE     40$                     ;MOVED ALL CHAR'S? (Y,N-40$)
2150  004744' 012602                       MOV     (SP)+,R2                ;GET # OF FILL SPACES
2151  004746' 001736                50$:    BEQ     CMVNEX                  ;ANY TO DO? (Y,N-CMVNEX)
2152  004750' 112721 000040                MOVB    #40,(R1)+               ;STORE A FILL SPACE
2153  004754' 005302                       DEC     R2                      ;DECR FILL CNT
2154  004756' 000773                       BR      50$                     ;GO CK IF DONE
2155
2156
2157  004760' 004567 001240        CSANAS: JSR     R5,CTYPE                ;REPORT NO "AS" ERROR
2158  004764' 000043                       .WORD   043
2159  004766' 002063                       .WORD   CERM21-.
2160  004770' 004567 001230        CSANNM: JSR     R5,CTYPE                ;REPORT "NO NAME" ERROR
2161  004774' 000042                       .WORD   042
2162  004776' 002043                       .WORD   CERM19-.
2163  005000' 004567 001220        CSAINM: JSR     R5,CTYPE                ;REPORT "INVALID NAME" ERROR
2164  005004' 000013                       .WORD   013
2165  005006' 002033                       .WORD   CERM19-.
2166  005010' 004567 001210        CSANPN: JSR     R5,CTYPE                ;ISSUE "NO PROG #" MSG
2167  005014' 000042                       .WORD   042
2168  005016' 001562                       .WORD   CERM01-.
```

```
                                        .SBTTL  /DELETE, /LIST, /ZERO, & /BOOT COMMANDS

                                ;       **********************
                                ;       *                    *
                                ;       *  /DELETE COMMAND   *
                                ;       *                    *
                                ;       **********************

005020' 005767  003170          CSLDEL: TST     CSCFWD                  ;ANOTHER PROG NAME?
005024' 100437                          BMI     CDLCEX                  ;Y,N-CDLCEX -- GO TO EXIT
005026' 004567  177614                  JSR     R5,CSACNR               ;SCAN PROG NAME & VALIDATE IT
005032' 004577  173006                  JSR     R5,@DELETE              ;GO DELETE THIS PROG
005036' 173330                          .WORD   CNCMND-.
005040' 000767                          BR      CSLDEL                  ;CHECK FOR ANOTHER NAME


                                ;       ********************
                                ;       *                  *
                                ;       *  /LIST COMMAND   *
                                ;       *                  *
                                ;       ********************

005042' 005046          CLST:   CLR     -(SP)                   ;RESET "ALL" & "FAST" FLAGS
005044' 005767  003144  2$:     TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
005050' 100417                  BMI     10$                     ;Y,N-10$
005052' 122777  000101  174402  CMPB    #'A,@CPNTR              ;IS IT "ALL"?
005060' 001002                  BNE     4$                      ;Y,N-4$
005062' 052716  000001          BIS     #1,(SP)                 ;SET THE "ALL" FLAG
005066' 122777  000106  174366  4$:     CMPB    #'F,@CPNTR      ;IS IT "FAST"?
005074' 001002                  BNE     6$                      ;Y,N-6$
005076' 052716  000400          BIS     #400,(SP)               ;SET THE "FAST" FLAG
005102' 004567  003114  6$:     JSR     R5,CSCN                 ;SCAN OVER THE OPERAND
005106' 000756                  BR      2$                      ;GO CK FOR ANOTHER OPERAND
005110' 012667  000004  10$:     MOV     (SP)+,20$               ;STORE THE "ALL" FLAG
005114' 004577  172730          JSR     R5,@LIST                ;GO LIST FETCH DEV DIRECTORY
005120' 000000  20$:     .WORD   XXXX
005122' 173244                  .WORD   CNCMND-.
005124' 000167  173236  CDLCEX: JMP     CNCMND                  ;RETURN FOR NEXT CMND


                                ;       ********************
                                ;       *                  *
                                ;       *  /ZERO COMMAND   *
                                ;       *                  *
                                ;       ********************

005130' 016700  172670  CZERO:  MOV     CSAVEI,R0               ;GET ADR OF SAVE DEV INFO
005134' 062700  000004          ADD     #4,R0                   ;POINT AT CURRENT DEV
005140' 012001                  MOV     (R0)+,R1                ;GET ITS MDL CODE
005142' 011003                  MOV     (R0),R3                 ;GET ITS UNIT #
005144' 004567  006376  20$:     JSR     R5,FDVNAM               ;FIND ITS DEV NAME & PUT
005150' 002744                  .WORD   CZWMSG+2-.              ;INTO THE MSG
005152' 105703                  TSTB    R3                      ;VALID UNIT #?
005154' 100004                  BPL     25$                     ;N,Y-25$
005156' 012767  020040  002734  MOV     #20040,CZWMSG+6         ;WIPE OUT COMMA & UNIT # IN MSG
005164' 000407                  BR      30$                     ;GO TO TYPE
```

```
2226  005166' 052703  000060          25$:   BIS     #60,R3                  ;MAKE UNIT # ASCII
2227  005172' 112767  000054  002720         MOVB    #',,CZWMSG+6            ;MOVE A COMMA TO THE MSG
2228  005200' 110367  002715                 MOVB    R3,CZWMSG+7             ;STORE UNIT # IN MSG
2229  005204' 004567  001014          30$:   JSR     R5,CTYPE                ;ISSUE WARNING MSG
2230  005210' 000420                         .WORD   420
2231  005212' 002700                         .WORD   CZWMSG-.
2232  005214' 000041                         .WORD   33.
2233  005216' 004577  172606                 JSR     R5,@CTRD                ;ISSUE READ FOR THE REPLY
2234  005222' 174242                         .WORD   CCMDRD-.
2235  005224' 000036                         .WORD   30.
2236  005226' 004567  002764                 JSR     R5,CSCNSP               ;SCAN FOR THE REPLY
2237  005232' 032702  000010                 BIT     #CSCFND,R2              ;ANY DATA ENTERED?
2238  005236' 001762                         BEQ     30$                     ;Y,N-30$
2239  005240' 122777  000116  174330         CMPB    #'N,@CDSTAD             ;WAS REPLY AN "N"?
2240  005246' 001726                         BEQ     CDLCEX                  ;N,Y-CDLCEX
2241  005250' 122777  000131  174320         CMPB    #'Y,@CDSTAD             ;WAS REPLY A "Y"?
2242  005256' 001352                         BNE     30$                     ;Y,N-30$
2243  005260' 004577  172574                 JSR     R5,@ZERO                ;GO TO ZERO DIRECTORY ROUTINE
2244  005264' 173102                         .WORD   CNCMND-.
2245  005266' 000716                         BR      CDLCEX                  ;RETURN FOR NXT CMND
2246
2247
2248                                  ;      ********************
2249                                  ;      *                  *
2250                                  ;      *  /BOOT COMMAND   *
2251                                  ;      *                  *
2252                                  ;      ********************
2253
2254  005270' 004577  172512          CBOOT: JSR     R5,@BOOT                ;GO TO EXEC'S BOOT ROUTINE
2255  005274' 173072                         .WORD   CNCMND-.
2256  005276' 000712                         BR      CDLCEX                  ;RET FOR NXT CMND IF ABORT
```

```
                                        .SBTTL  ASSIGN COMMAND ROUTINE

                                ;       ********************
                                ;       *                  *
                                ;       *  ASSIGN COMMAND  *
                                ;       *                  *
                                ;       ********************
                                ;
005300' 004567  005254          CASSIG: JSR     R5,CVLMD1               ;GO VALIDATE MDL NAME & DEV #'S
005304' 016701  174310                  MOV     CMCDVN,R1               ;GET MODEL CODE
005310' 005767  002700                  TST     CSCFWD                  ;ANOTHER OPERAND IN CMND?
005314' 100635                          BMI     CSANPN                  ;Y,N-CSANPN
005316' 016700  174140                  MOV     CPNTR,R0                ;GET ADR OF NXT OPER FIRST CHAR
005322' 122710  000106                  CMPB    #'F,(R0)                ;FIRST CHAR AN "F"?
005326' 001443                          BEQ     CASFTH                  ;N,Y-CASFTH
005330' 122710  000123                  CMPB    #'S,(R0)                ;IS IT AN "S"?
005334' 001446                          BEQ     CASAVE                  ;N,Y-CASAVE
005336' 122710  000114                  CMPB    #'L,(R0)                ;AN "L"?
005342' 001454                          BEQ     CASLST                  ;N,Y-CASLST
005344' 005767  026644                  TST     CSYSFW                  ;CONSOLE REQ BEING PROCESSED?
005350' 100002                          BPL     10$                     ;Y,N-10$
005352' 000167  173376                  JMP     CRSCMD                  ;GO REPORT 'RESTRICTED CMND' ERROR
005356' 004567  003122          10$:    JSR     R5,CVPNUM               ;GO CHECK IT FOR A PROG #
                   001                  .IF DF MM
                                        JSR     PC,UPAPAR               ;MAP PAR'S TO USER'S PROG AREA
                   000                  .ENDC
005362' 005703                          TST     R3                      ;PROG EXIST FOR THIS #?
005364' 001002                          BNE     20$                     ;N,Y-20$
005366' 000167  174346                  JMP     CCFPDE                  ;REPORT PROG DOES NOT EXIST ERROR
005372' 126367  000033  174221  20$:    CMPB    PMDLCD+1(R3),CMCDVN+1   ;SAME DEVICE ROUTINE?
005400' 001123                          BNE     CASER2                  ;Y,N-CASER2
005402' 004567  006006                  JSR     R5,CSTDVI               ;STORE DEV INFO IN PROG TBL
005406' 042713  100000                  BIC     #ACTIVE,(R3)            ;RESET PROG'S ACTIVE FLAG
005412' 122767  000377  174200          CMPB    #377,CMCDVN             ;IS IT THE "NONE" DEV ROUT?
005420' 001504                          BEQ     CASEX                   ;N,Y-CASEX
005422' 010304                          MOV     R3,R4                   ;SET UP ADR OF DEV REG &
005424' 062704  000266                  ADD     #PTLGTH+DEVDRA,R4       ;INT VECT INFO
005430' 004567  003536                  JSR     R5,CRQADR               ;ALLOW DEVICE INFO CHANGE IF NEEDED
005434' 000476                          BR      CASEX                   ;RETURN FOR NEXT CMND

                                ;       FETCH ASSIGN

005436' 032701  000004          CASFTH: BIT     #04,R1                  ;VALID FETCH DEV?
005442' 001476                  CASE1J: BEQ     CASER1                  ;Y,N-CASER1
005444' 016702  172342                  MOV     CFTCHI,R2               ;SET UP FETCH DEV INFO ADR
005450' 000405                          BR      CASFCM                  ;GO TO SAV/FTH COMMON

                                ;       SAVE ASSIGN

005452' 032701  000002          CASAVE: BIT     #02,R1                  ;VALID SAVE DEV?
005456' 001470                          BEQ     CASER1                  ;Y,N-CASER1
005460' 016702  172340                  MOV     CSAVEI,R2               ;SET UP SAVE DEV INFO ADR
005464' 005046                  CASFCM: CLR     -(SP)                   ;RESET REQ ADR'S FLAG
005466' 016746  035126                  MOV     CLDID+4,-(SP)           ;GET "LOAD" MDL CODE
005472' 000411                          BR      CASCOM                  ;GO TO COMMON PROCESSING
```

```
2314                                        ;       LIST ASSIGN
2315
2316  005474' 032701  000001        CASLST: BIT     #01,R1          ;VALID LIST DEV?
2317  00550J' 001457                        BEQ     CASER1          ;Y,N-CASER1
2318  005502' 016702  172306                MOV     CLISTI,R2       ;SET UP LIST DEV INFO ADR
2319  005506' 012746  000001                MOV     #1,-(SP)        ;SET REQ ADR'S FLAG
2320  005512' 016746  035066                MOV     CKBDID+4,-(SP)  ;GET "KYBD/KB00" MDL CODE
2321
2322  005516' 020126                CASCOM: CMP     R1,(SP)+        ;ACCEPTABLE DEFAULT NAME?
2323  005520' 001403                        BEQ     60$             ;N,Y-60$
2324  005522' 020167  035050                CMP     R1,CNONID+4     ;IS THIS A 'NONE' ASSIGN?
2325  005526' 001017                        BNE     70$             ;Y,N-70$
2326  005530' 026702  172260        60$:    CMP     CLISTI,R2       ;IS THIS FOR THE PRINTER?
2327  005534' 001006                        BNE     65$             ;Y,N-65$
2328  005536' 032767  004000  026450        BIT     #CHPINT,CSYSFW  ;INT ENABLE BEING HELD SET?
2329  005544' 001402                        BEQ     65$             ;Y,N-65$
2330  005546' 005072  000010                CLR     @8.(R2)         ;RESET PRINTER'S INT ENB
2331  005552' 012200                65$:    MOV     (R2)+,R0        ;GET DEFAULT MDL CODE
2332  005554' 012201                        MOV     (R2)+,R1        ;& UNIT #
2333  005556' 010022                        MOV     R0,(R2)+        ;STORE THEM AS CURRENT
2334  005560' 110112                        MOVB    R1,(R2)         ;VALUES
2335  005562' 005726                        TST     (SP)+           ;CLEAN UP STACK
2336  005564' 000422                        BR      CASEX           ;GO TO CMND EXIT
2337  005566' 122701  000377        70$:    CMPB    #377,R1         ;THIS DEFAULT TYPE OF DEV?
2338  005572' 001421                        BEQ     CASER0          ;N,Y-CASER0
2339  005574' 042701  000360                BIC     #360,R1         ;RESET DIFFERENTIATOR BITS IN MDL CD
2340  005600' 062702  000004                ADD     #4,R2           ;POINT AT CURR DEV WORDS
2341  005604' 010122                        MOV     R1,(R2)+        ;STORE DEV'S MDL CODE WORD
2342  005606' 116712  174011                MOVB    CMCDVN+3,(R2)   ;STORE FIRST DEV #
2343  005612' 005726                        TST     (SP)+           ;REQUEST DEV ADR'S?
2344  005614' 001406                        BEQ     CASEX           ;Y,N-CASEX
2345  005616' 005722                        TST     (R2)+           ;POINT AT DEVICE INFO
2346  005620' 010204                        MOV     R2,R4           ;SET DEV INFO ADR FOR S/R
2347  005622' 004567  002374                JSR     R5,CSCN         ;SCAN LAST WORD & CK REPLY TERMINATOR
2348  005626' 004567  003340                JSR     R5,CRQADR       ;GO REQUEST DEV ADR'S
2349  005632' 000167  172530        CASEX:  JMP     CNCMND          ;RETURN FOR NEXT CMND
2350
2351  005636' 005726                CASER0: TST     (SP)+           ;TAKE REQ ADR FLG OFF STACK
2352  005640' 004567  000360        CASER1: JSR     R5,CTYPE        ;GO REPORT INV DEV TYPE
2353  005644' 000012                        .WORD   012
2354  005646' 001210                        .WORD   CERM22-.
2355  005650' 004567  000350        CASER2: JSR     R5,CTYPE        ;TELL HIM HE CAN'T
2356  005654' 000002                        .WORD   002             ;CHANGE DEVICES
2357  005656' 001331                        .WORD   CERM31-.
```

```
                                        .SBTTL  RDM & WRM COMMAND ROUTINES

                                ;       ******************
                                ;       *                  *
                                ;       *  WRM COMMAND     *
                                ;       *                  *
                                ;       ******************

005660' 005767  002330          CWRM:   TST     CSCFWD          ;IS THERE A DATA OPERAND?
005664' 100002                  BPL     CWRMOK                  ;N,Y-CWRMOK
005666' 000167  174136          CN2OPJ: JMP     CN2OER          ;GO REPORT NO SECOND OPER ERROR
005672' 004567  005546          CWRMOK: JSR     R5,CMEMCK       ;GO CK FOR OUT OF MEM ADR
005676' 004567  002332                  JSR     R5,CSCNCB       ;GO CONVERT NXT OPER TO BINARY
            001                         .IF DF MM
                                        .IFF
005702' 010024                          MOV     R0,(R4)+        ;WRITE WORD TO MEMORY
                                        .IFT
                                        MOV     R0,-(SP)        ;SAVE DATA WORD
                                        MOV     R4,R0           ;MOVE ADR TO CORRECT REGS
                                        MOV     R3,R2
                                        JSR     PC,CVTVAD       ;CONVERT IT TO A VIRTUAL ADR
                                        MOV     (SP)+,(R1)      ;WRITE WORD TO MEMORY
                                        ADD     #2,R4           ;POINT TO NXT MEM LOCATION
                                        ADC     R3
            000                         .ENDC
005704' 005767  002304                  TST     CSCFWD          ;STILL ANOTHER OPERAND?
005710' 100370                  BPL     CWRMOK                  ;N,Y-CWRMOK
005712' 000167  172450          JMP     CNCMND                  ;RETURN FOR ANOTHER CMND

                                ;       ******************
                                ;       *                  *
                                ;       *  RDM COMMAND     *
                                ;       *                  *
                                ;       ******************

            001                         .IF DF MM
                                        .IFF
005716' 010405                  CRDM:   MOV     R4,R5           ;SAVE FIRST ADR AS START & END ADR
                                        .IFT
                                CRDM:   MOV     #1,R5           ;SET CNT TO 1 WORD
                                        .IFTF
005720' 005767  002270                  TST     CSCFWD          ;IS THERE A SECOND OPERAND?
005724' 100405                  BMI     40$                     ;Y,N-40$
005726' 004567  002302          JSR     R5,CSCNB8               ;GO SCAN & CONVERT IT TO BINARY
005732' 004567  005406          JSR     R5,CKODAD               ;GO CHECK FOR ODD ADR
                                        .IFF
005736' 010005                  MOV     R0,R5                   ;SET ADR UP AS END ADR
005740' 010400                  40$:    MOV     R4,R0           ;SET UP CURR MEM ADR
                                        .IFT
                                        SUB     R4,R0           ;COMPUTE # OF BYTES
                                        SBC     R2
                                        SUB     R3,R2
                                        BMI     40$             ;NEG BYTE CNT? (N,Y-40$)
                                        MOV     R0,R5           ;SAVE BYTE CNT
                                        ASR     R5              ;MAKE IT A WORD CNT
                                        INC     R5              ;ALLOW FOR FINAL WORD
```

```
2415                                        40$:    MOV     R4,R0                   ;SET CURRENT ADR IN
2416                                                MOV     R3,R2                   ;CORRECT REGS
2417                                                .IFTF
2418  005742' 004567  033154                        JSR     R5,BINAS8               ;CONVERT IT TO ASCII &
2419  005746' 001431                                .WORD   CRDMG1-.                ;PUT IN DISPLAY MSG
2420  005750' 004567  005470                        JSR     R5,CMEMCK               ;GO CK FOR OUT OF MEM ADR
2421                                                .IFF
2422  005754' 011400                                MOV     (R4),R0                 ;GET DATA AT CURR MEM ADR
2423                                                .IFT
2424                                                MOV     R4,R0                   ;MOVE ADR TO CORRECT REGS
2425                                                MOV     R3,R2
2426                                                JSR     PC,CVTVAD               ;CONVERT IT TO A VIRTUAL ADR
2427                                                MOV     (R1),R0                 ;GET THE DATA WORD
2428                                                .IFTF
2429  005756' 004567  033140                        JSR     R5,BINASC               ;CONVERT IT TO ASCII &
2430  005762' 001425                                .WORD   CRDMDT-.                ;PUT IN DISPLAY MSG
2431  005764' 004567  000234                        JSR     R5,CTYPE                ;ISSUE ADR & DATA DISPLAY
2432  005770' 000400                                .WORD   400                     ;MSG
2433  005772' 001405                                .WORD   CRDMG1-.
2434                                                .IFF
2435  005774' 020405                                CMP     R4,R5                   ;START ADR >= END ADR?
2436  005776' 103402                                BLO     50$                     ;Y,N-50$
2437                                                .IFT
2438                                                DEC     R5                      ;DECR WORD CNT
2439                                                BNE     50$                     ;CNT = 0?
2440                                                .IFTF
2441  006000' 000167  173756                        JMP     CCFMTB                  ;EXIT
2442                                                .IFF
2443  006004' 005724                50$:    TST     (R4)+                   ;INCR CURR MEM ADR
2444  006006' 020405                        CMP     R4,R5                   ;THIS LAST LOC TO DISPLAY?
2445  006010' 001753                        BEQ     40$                     ;N,Y-50$
2446                                        .IFT
2447                                50$:    ADD     #2,R4                   ;ADD 2 TO THE ADR
2448                                        ADC     R3
2449                                        CMP     R5,#1                   ;THIS LAST LOC TO DISPLAY?
2450                                        BEQ     40$                     ;N,Y-40$
2451                                        .IFTF
2452  006012' 032704  000007                BIT     #7,R4                   ;THIS ADR AT OCT 10 BOUNDARY?
2453  006016' 001750                        BEQ     40$                     ;N,Y-40$
2454                                        .IFF
2455  006020' 011400                        MOV     (R4),R0                 ;GET DATA AT THIS ADR
2456                                        .IFT
2457                                        MOV     R4,R0                   ;ADR TO CORRECT REGS
2458                                        MOV     R3,R2
2459                                        JSR     PC,CVTVAD               ;CONVERT IT TO A VIRTUAL ADR
2460                                        MOV     (R1),R0                 ;GET THE DATA WORD
2461                                        .IFTF
2462  006022' 004567  033074                JSR     R5,BINASC               ;CONVERT IT TO ASCII &
2463                                        .IFF
2464  006026' 001361                        .WORD   CRDMDT-.                ;PUT IN DATA MSG
2465                                        .IFT
2466                                        .WORD   CRDMD2-.
2467                                        .IFTF
2468  006030' 004567  000170                JSR     R5,CTYPE                ;DISPLAY DATA ONLY
2469  006034' 000400                        .WORD   400
2470  006036' 001350                        .WORD   CRDMG2-.
```

```
2471                                    .IFT
2472                                    DEC     R5                      ;DECR WORD COUNT
2473                                    .IFTF
2474  006040' 000761                   BR      50$                     ;GO PROCESS NEXT WORD
2475                  000               .ENDC
```

```
                                        .SBTTL  ADD, SUB, BOC, CDB, & CBD COMMANDS


                                ;       ******************
                                ;       *                *
                                ;       *  ADD COMMAND   *
                                ;       *                *
                                ;       ******************

006042' 005004          CADD:   CLR     R4              ;CLEAR ACCUMULATOR REG
        001                     .IF DF MM
                                CLR     R3
        000                     .ENDC
006044' 005767  002144  10$:    TST     CSCFWD          ;IS THERE ANOTHER OPERAND?
006050' 100404                  BMI     CARITH          ;Y,N-CARITH
006052' 004567  002156          JSR     R5,CSCNB8       ;SCAN & CONVERT IT
006056' 060004                  ADD     R0,R4           ;ADD IT TO RUNNING TOTAL
        001                     .IF DF MM
                                ADC     R3
                                ADD     R2,R3
        000                     .ENDC
006060' 000771                  BR      10$             ;GO CK FOR ANOTHER OPERAND
006062' 010400          CARITH: MOV     R4,R0           ;SET UP ANS FOR S/R
        001                     .IF DF MM
                                MOV     R3,R2
        000                     .ENDC
006064' 004567  033032  CARIT1: JSR     R5,BINAS8       ;CONVERT TO ASCII
006070' 001634                  .WORD   CANS-.
006072' 004567  000126  CARDSP: JSR     R5,CTYPE        ;ISSUE ANSWER MSG
006076' 000400                  .WORD   400
006100' 001616                  .WORD   CANSMG-.
006102' 000167  172260  CARRET: JMP     CNCMND          ;RET FOR NXT CMND


                                ;       **********************
                                ;       *                    *
                                ;       *  SUBTRACT COMMAND  *
                                ;       *                    *
                                ;       **********************

006106' 004567  002122  CSUB:   JSR     R5,CSCNB8       ;SCAN & CONVERT SUBTRAHEND
006112' 010004                  MOV     R0,R4           ;SAVE IT
        001                     .IF DF MM
                                .IFT
                                MOV     R2,R3
                                .IFTF
006114' 005000                  CLR     R0              ;CLEAR MINUEND REG
                                .IFT
                                CLR     R2
        000                     .ENDC
006116' 005767  002072          TST     CSCFWD          ;IS THERE A MINUEND?
006122' 100402                  BMI     30$             ;Y,N-30$
006124' 004567  002104          JSR     R5,CSCNB8       ;SCAN & CONVERT IT
006130' 160400          30$:    SUB     R4,R0           ;SUBTRACT TWO NUMBERS
        001                     .IF DF MM
                                SBC     R2
```

```
2533                                    SUB     R3,R2
2534                    000             .ENDC
2535  006132' 000754                   BR      CARIT1                  ;GO DISPLAY ANSWER
```

```
2537                                    ;       ****************************************
2538                                    ;       *                                      *
2539                                    ;       *  BRANCH OFFSET CALCULATION COMMAND   *
2540                                    ;       *                                      *
2541                                    ;       ****************************************
2542
2543  006134' 005767  002054            CBOC:   TST     CSCFWD                  ;IS THERE A DEST ADR?
2544  006140' 100652                            BMI     CN2OPJ                  ;Y,N-CN2OPJ
2545  006142' 062704  000002                    ADD     #2,R4                   ;POINT INST ADR AT NXT INST
2546             001                            .IF DF MM
2547                                            ADC     R3
2548             000                            .ENDC
2549  006146' 004567  002062                    JSR     R5,CSCNBB               ;CONVERT DEST ADR TO BINARY
2550  006152' 004567  005166                    JSR     R5,CKODAD               ;CK FOR AN ODD ADR
2551  006156' 160400                            SUB     R4,R0                   ;SUB INST ADR FROM DEST ADR
2552             001                            .IF DF MM
2553                                            .IFT
2554                                            SBC     R2
2555                                            SUB     R3,R2
2556                                            ASR     R2                      ;DIVIDE ANS BY 2
2557                                            ROR     R0
2558                                            .IFF
2559  006160' 006200                            ASR     R0                      ;DIVIDE ANS BY 2
2560             000                            .ENDC
2561  006162' 000740                            BR      CARIT1                  ;GO DISPLAY THE ANSWER
2562
2563
2564                                    ;       ********************************
2565                                    ;       *                              *
2566                                    ;       *  CONVERT DECIMAL TO BINARY   *
2567                                    ;       *                              *
2568                                    ;       ********************************
2569
2570  006164' 004567  002052            CCDB:   JSR     R5,CSCNCD               ;SCAN & CONVERT TO PKED DEC
2571  006170' 004567  033722                    JSR     R5,DECBIN               ;CONVERT IT TO BINARY
2572  006174' 000004                            .WORD   CCDBER-.
2573             001                            .IF DF MM
2574                                            .IFF
2575  006176' 000732                            BR      CARIT1                  ;GO DISPLAY ANSWER
2576                                            .IFT
2577                                            JSR     R5,BINASC               ;CONVERT IT TO ASCII
2578                                            .WORD   CCDBAN-.
2579                                    CCDBDS: JSR     R5,CTYPE                ;ISSUE CONV ANSWER MSG
2580                                            .WORD   400
2581                                            .WORD   CCDBMG-.
2582                                            BR      CARRET                  ;RETURN FOR NEXT CMND
2583             000                            .ENDC
2584
2585  006200' 004567  000020            CCDBER: JSR     R5,CTYPE                ;GO ISSUE INV NUM MSG
2586  006204' 000012                            .WORD   012
2587  006206' 000724                            .WORD   CERM24-.
2588
2589
2590                                    ;       ********************************
2591                                    ;       *                              *
2592                                    ;       *  CONVERT BINARY TO DECIMAL   *
```

```
2593                                    ;       *                                 *
2594                                    ;       ***********************************
2595
2596  006210' 004567  002020           CCBD:   JSR     R5,CSCNCB               ;SCAN & CONVERT # TO BINARY
2597  006214' 004567  032752                   JSR     R5,BTASLZ               ;CONVERT BINARY # TO DECIMAL ASCII
2598             001                            .IF DF MM
2599                                            .IFF
2600  006220' 001504                           .WORD   CANS-.
2601  006222' 000723                           BR      CARDSP                  ;GO DISPLAY ANSWER
2602                                            .IFT
2603                                           .WORD   CCDBAN-.
2604                                           BR      CCDBDS                  ;GO DISPLAY ANSWER
2605             000                            .ENDC
```

```
                        .SBTTL  CONSOLE TERMINAL MESSAGE FORMATTING ROUTINE
;*****************************************************************************
;
;  THIS ROUTINE FORMATS ALL MESSAGES ISSUED TO THE CONSOLE
;  TERMINAL. IT PROVIDES SEVERAL FUNCTIONS WHICH ARE CONTROLLED
;  BY BITS IN THE SUPPLIED FUNCTION WORD. OPTIONS INCLUDE
;  REMOVING OF A WORD FROM THE STACK, PRECEDING THE MSG WITH '*ER* ',
;  PRECEDING THE MSG WITH 'PROG # XX', COMPUTING THE MSG LENGTH OR
;  USING A SUPPLIED COUNT, TYPING OF AN EXTRA CR/LF, DISPLAYING OF THE
;  LAST WORD SCANNED (INVALID DATA) IN THE COMMAND READ-IN AREA, AND EITHER
;  EXITING IN-LINE OR TO AN ADDRESS STORED AT A STANDARD LOCATION.
;  WHEN COMPUTING THE CHAR COUNT, TERMINATION WILL OCCUR ON THE
;  FIRST 00 CHAR. THIS CHAR WILL NOT BE INCLUDED IN THE DATA SENT TO
;  THE TERMINAL. A CR/LF WILL BE ISSUED INSTEAD.
;
;  LINKAGE:     JSR     R5,CTYPE        ROUTINE CALL
;               .WORD   FUNCWD          FUNCTION WORD
;               .WORD   MSGADR-.        RELATIVE ADR OF MSG
;               .WORD   CHARCNT         OPTIONAL MSG CHAR CNT; THIS
;                                       IS REQUIRED WHEN 2--4 OF THE
;                                       FUNCTION WORD IS SET.
;
;  FUNCTION WORD FORMAT:
;
;       2--0 = 1 =      REMOVE WORD BEFORE THIS JSR'S R5 FROM STACK
;
;       2--1 = 1 =      PRECEDE MSG WITH '*ER* '
;
;       2--2 = 1 =      PRECEDE MSG WITH 'PROG # XX ' AND TAILOR
;                       XX TO THE CURRENT PROGRAM NUMBER
;
;       2--3 = 1 =      PRECEDE MSG WITH 'INV '
;
;       2--4 = 1 =      DO NOT COMPUTE MSG LENGTH, USE SUPPLIED
;                       COUNT INSTEAD
;
;       2--5 = 1 =      TYPE ' NOT SPECIFIED' AFTER THE MSG
;
;       2--6 = 1 =      DISPLAY INVALID COMMAND DATA
;
;       2--7 = 1 =      TYPE AN EXTRA CR/LF AFTER THE MSG
;
;       2--8 = 1 =      EXIT IN-LINE
;            = 0 =      EXIT TO ADDRESS STORED AT "CERRET"
;
;  BIT LAYOUT:
;       EXIT IN-LINE        TYPE 'NOT SPEC'     TYPE 'PROG # XX'
;       TYPE XTRA CR/LF     USE CHAR CNT        TYPE '*ER* '
;       DISP INV DATA       TYPE 'INV '         CLEAR STACK
;
;  DESTROYS REGISTERS: R0,R1,R2
;
;  NOTE:        WHEN 2--2 OF THE FUNCTION WORD IS SET (TYPE PRG #),
;               R4 MUST CONTAIN THE ADR OF THE CURRENT CONTROL
;               ROUTINE TABLE ENTRY. IT WILL BE LEFT UNCHANGED.
;
;*****************************************************************************
```

```
2663
2664
2665  006224' 032715 000001          CTYPE:  BIT     #1,(R5)                 ;CLEAR A WORD OFF STACK?
2666  006230' 001401                         BEQ     10$                     ;Y,N-10$
2667  006232' 012616                         MOV     (SP)+,(SP)              ;REMOVE WORD BEFORE RETURN ADR
2668  006234' 012546                 10$:    MOV     (R5)+,-(SP)             ;SAVE ROUTINE'S FUNCTION WORD
2669  006236' 032716 000002                  BIT     #2,(SP)                 ;ISSUE *ER* AT BEGINNING OF LINE?
2670  006242' 001407                         BEQ     20$                     ;Y,N-20$
2671  006244' 042767 010000 025742          BIC     #CNTRLO,CSYSFW          ;RESET SUPPRESS TYPING FLG IF SET
2672  006252' 004577 171562                  JSR     R5,@CTWR                ;TYPE THE *ER*
2673  006256' 001102                         .WORD   CERID-.
2674  006260' 000005                         .WORD   5.
2675  006262' 032716 000004          20$:    BIT     #4,(SP)                 ;ISSUE 'PROG # XX' DATA?
2676  006266' 001410                         BEQ     30$                     ;Y,N-30$
2677  006270' 004767 005314                  JSR     PC,CPNASC               ;COMPUTE PROG # IN ASCII
2678  006274' 010167 001074                  MOV     R1,CEPNUM               ;MOVE   PROG # TO MSG
2679  006300' 004577 171534                  JSR     R5,@CTWR                ;TYPE 'PROG # XX ' DATA
2680  006304' 001061                         .WORD   CPNMSG-.
2681  006306' 000012                         .WORD   10.
2682  006310' 032716 000010          30$:    BIT     #10,(SP)                ;ISSUE 'INV ' AT BEGINNING OF MSG?
2683  006314' 001404                         BEQ     33$                     ;Y,N-33$
2684  006316' 004577 171516                  JSR     R5,@CTWR                ;TYPE 'INV ' DATA
2685  006322' 000721                         .WORD   CIVMSG-.
2686  006324' 000004                         .WORD   4
2687  006326' 010500                 33$:    MOV     R5,R0                   ;GET MSG START ADR AND
2688  006330' 062500                         ADD     (R5)+,R0                ;MAKE IT ABSOLUTE
2689  006332' 010001                         MOV     R0,R1                   ;SAVE START ADR
2690  006334' 010702                         MOV     PC,R2                   ;MAKE IT RELATIVE TO THIS
2691  006336' 062702 000042                  ADD     #50$-.,R2               ;ROUTINE AND STORE IN CTWR
2692  006342' 160200                         SUB     R2,R0                   ;LINK INFO
2693  006344' 010022                         MOV     R0,(R2)+
2694  006346' 032716 000020                  BIT     #20,(SP)                ;MSG CHAR CNT SUPPLIED?
2695  006352' 001402                         BEQ     35$                     ;Y,N-35$
2696  006354' 012512                         MOV     (R5)+,(R2)              ;STORE SUPPLIED CNT IN LINK INFO
2697  006356' 000406                         BR      45$                     ;GO ISSUE MSG
2698  006360' 005000                 35$:    CLR     R0                      ;CLEAR CHAR CNT REG
2699  006362' 005200                 40$:    INC     R0                      ;INCR CHAR CNT BY 1
2700  006364' 105721                         TSTB    (R1)+                   ;IS MSG CHAR A 00?
2701  006366' 001375                         BNE     40$                     ;Y,N-40$
2702  006370' 005300                         DEC     R0                      ;DON'T COUNT THE 00 BYTE
2703  006372' 010012                         MOV     R0,(R2)                 ;STORE COMPUTED CNT IN LINK INFO
2704  006374' 004577 171440          45$:    JSR     R5,@CTWR                ;ISSUE MSG SPECIFIED
2705  006400' 000000                 50$:    .WORD   XXXX
2706  006402' 000000                         .WORD   XXXX
2707  006404' 032716 000040                  BIT     #40,(SP)                ;ISSUE ' NOT SPECIFIED' AFTER THE MSG?
2708  006410' 001404                         BEQ     53$                     ;Y,N-53$
2709  006412' 004577 171422                  JSR     R5,@CTWR                ;TYPE ' NOT SPECIFIED' DATA
2710  006416' 000631                         .WORD   CNSMSG-.
2711  006420' 000016                         .WORD   14.
2712  006422' 032716 000020          53$:    BIT     #20,(SP)                ;WAS CHAR CNT SUPPLIED?
2713  006426' 001004                         BNE     56$                     ;N,Y-56$
2714  006430' 004577 171404                  JSR     R5,@CTWR                ;ISSUE A CR/LF
2715  006434' 000660                         .WORD   CECMSG-.
2716  006436' 000002                         .WORD   2
2717  006440' 032716 000100          56$:    BIT     #100,(SP)               ;ISSUE INVALID DATA MSG?
2718  006444' 001437                         BEQ     90$                     ;Y,N-90$
```

```
2719  006446' 016700 173124          MOV     CDSTAD,R0       ;GET ADR OF CMND WORD
2720  006452' 016701 173122          MOV     CDCNT,R1        ;GET # OF CHARS IN WORD
2721  006456' 012702 000014          MOV     #12.,R2         ;SET UP MAX COUNT OF DEC 12
2722  006462' 020102                 CMP     R1,R2           ;WORD LENGTH > 12?
2723  006464' 101401                 BLOS    60$             ;Y,N-60$
2724  006466' 010201                 MOV     R2,R1           ;SET WORD LENGTH TC 12
2725  006470' 010546          60$:   MOV     R5,-(SP)        ;SAVE WORK REG
2726  006472' 010705                 MOV     PC,R5           ;GET MSG STORAGE ADR
2727  006474' 062705 000604          ADD     #CERDTA-.,R5
2728  006500' 005701                 TST     R1              ;IS WORD LENGTH = 0?
2729  006502' 001404                 BEQ     70$             ;N,Y-70$
2730  006504' 112025          65$:   MOVB    (R0)+,(R5)+     ;MOVE WORD CHAR TO ERROR MSG
2731  006506' 005302                 DEC     R2              ;DECR MAX COUNT
2732  006510' 005301                 DEC     R1              ;DECR WORD CNT
2733  006512' 001374                 BNE     65$             ;WD CNT = 0? (Y,N-65$)
2734  006514' 005702          70$:   TST     R2              ;REMAINING MAX CNT = 0?
2735  006516' 001405                 BEQ     80$             ;N,Y-80$
2736  006520' 012701 000040          MOV     #40,R1          ;SET UP SPACE CHAR
2737  006524' 110125          75$:   MOVB    R1,(R5)+        ;FILL AFTER DATA WITH A SPACE
2738  006526' 005302                 DEC     R2              ;DECR MAX CNT
2739  006530' 001375                 BNE     75$             ;MAX CNT = 0? (Y,N-75$)
2740  006532' 012605          80$:   MOV     (SP)+,R5        ;RESTORE WORK REG
2741  006534' 004577 171300          JSR     R5,@CTWR        ;DISPLAY INVALID DATA
2742  006540' 000525                 .WORD   CERMSG-.
2743  006542' 000031                 .WORD   25.
2744  006544' 032716 000200  90$:    BIT     #200,(SP)       ;ISSUE EXTRA CR/LF?
2745  006550' 001404                 BEQ     95$             ;Y,N-95$
2746  006552' 004577 171262          JSR     R5,@CTWR        ;TYPE A CR/LF
2747  006556' 000536                 .WORD   CECMSG-.
2748  006560' 000002                 .WORD   2
2749  006562' 032726 000400  95$:    BIT     #400,(SP)+      ;EXIT IN-LINE?
2750  006566' 001401                 BEQ     100$            ;Y,N-100$
2751  006570' 000205                 RTS     R5              ;EXIT IN-LINE
2752
2753  006572' 012605          100$:  MOV     (SP)+,R5        ;CLEAR REG R5 OFF THE STACK
2754  006574' 000177 173002          JMP     @CERRET         ;EXIT TO PRE-STORED RETURN ADR
```

```
2756
2757                                    ;       ****    MESSAGE STORAGE AREA  ****
2758
2759
2760
2761                                                    .NLIST  BEX
2762
2763  006600' 051120  043517  021440  CERM01: .ASCIZ  /PROG #/
2764  006607'    104  052101  020101  CERM02: .ASCIZ  /DATA CONV ERR/
2765  006625'    122  051505  051124  CERM03: .ASCII  /RESTRICTED /
2766  006640' 046503  042116     000  CERM04: .ASCIZ  /CMND/
2767  006645'    104  042517  047123  CERM05: .ASCIZ  /DOESN'T EXIST/
2768  006663'    125  047116  041505  CERM06: .ASCII  /UNNEC /
2769  006671'    117  042520  040522  CERM07: .ASCIZ  /OPERAND/
2770  006701'     062  042116  047440  CERM08: .ASCIZ  /2ND OPERAND/
2771  006715'    101  051104  037040  CERM09: .ASCIZ  /ADR > MEM/
2772  006727'    102  043125  041457  CERM10: .ASCIZ  'BUF/COM'
2773  006737'    102  043125  041457  CERM11: .ASCIZ  'BUF/COM #'
2774  006751'     047  044527  044124  CERM12: .ASCIZ  /'WITH'/
2775  006760' 040504  040524     000  CERM13: .ASCIZ  /DATA/
2776  006765'    116  020117  051120  CERM14: .ASCIZ  /NO PROG'S FND/
2777  007003'    116  020117  041117  CERM15: .ASCIZ  /NO OBJ CODE/
2778  007017'    114  047111  020105  CERM16: .ASCIZ  /LINE #/
2779  007026' 047125  052111  021440  CERM17: .ASCIZ  /UNIT #/
2780  007035'    115  046104     040  CERM18: .ASCII  /MDL /
2781  007041'    116  046501  000105  CERM19: .ASCIZ  /NAME/
2782  007046' 051102     000          CERM20: .ASCIZ  /BR/
2783  007051'     047  051501  000047  CERM21: .ASCIZ  /'AS'/
2784  007056' 042504  020126  054524  CERM22: .ASCIZ  /DEV TYPE/
2785  007067'    111  051516  043125  CERM23: .ASCIZ  'INSUF MEM FOR DEV ROUT & I/O AREAS'
2786  007132' 052516  041115  051105  CERM24: .ASCIZ  /NUMBER/
2787  007141'    117  042104  040440  CERM25: .ASCIZ  /ODD ADR/
2788  007151'    101  051104  051040  CERM26: .ASCIZ  /ADR RANGE/
2789  007163'    111  040516  052103  CERM27: .ASCIZ  /INACTIVE/
2790  007174' 047111  041440  047117  CERM30: .ASCIZ  /IN CONTROL/
2791  007207'    103  047101  052047  CERM31: .ASCIZ  /CAN'T CHANGE DEV TYPE/
2792  007235'    120  052101  021440  CERM32: .ASCIZ  /PAT #/
2793               001                          .IF DF MM
2794                                    CERM33: .ASCIZ  /RDIO+WRIO > 24,250/
2795                                    CERM34: .ASCIZ  /PROG AREA IN USE/
2796                                    CERM35: .ASCIZ  /ADR WITHIN MPG/
2797                                    CERM36: .ASCIZ  /'TO'/
2798                                    CERM37: .ASCIZ  /DESTINATION/
2799                                    CERM38: .ASCIZ  /INSUF ROOM FOR PROG/
2800                                    CERM39: .ASCIZ  /REG # (SHD BE 7 - 28,29,30)/
2801                                    CERM40: .ASCIZ  /INSUF REGS AVAIL/
2802                                    CERM41: .ASCIZ  /REGS IN USE/
2803                                    CERM42: .ASCIZ  'WARNING!! PROG I/O AREA > 18 BIT ADR - MAY NOT RUN'
2804               000                          .ENDC
2805
2806
2807
2808  007243'    111  053116     040  CIVMSG: .ASCII  /INV /
2809  007247'     040  047516  020124  CNSMSG: .ASCII  / NOT SPECIFIED/
2810  007265'    111  053116  042040  CERMSG: .ASCII  /INV DATA = /           ;MUST PRECEDE CECMSG
2811  007300' 000014                   CERDTA: .BLKB   14
```

```
2812  007314' 005015  047105  042524  CECMSG: .ASCIZ  <015><012>/ENTER CMND/
2813  007331'    015  025012     052  CASMSG: .ASCII  <015><012>/**/
2814               001                        .IF DF MM
2815                                          .IFF
2816  007335'    130  054130  054130  CAAMSG: .ASCIZ  /XXXXXX ADJ TO EVEN/
2817                                          .IFT
2818                                  CAAMSG: .ASCIZ  /XXXXXXXX ADJ TO EVEN/
2819               000                        .ENDC
2820                                          .EVEN
2821  007360' 042452  025122     040  CERID:  .ASCII  /*ER* /
2822  007365'    120  047522  020107  CPNMSG: .ASCII  /PROG # /
2823  007374' 054130     040          CEPNUM: .ASCII  /XX /   ;WD BNDRY
2824               001                        .IF DF MM
2825                                          .IFF
2826  007377'    130  054130  054130  CRDMG1: .ASCII  /XXXXXX=/
2827  007406'    011                  CRDMG2: .BYTE   011
2828  007407'    130  054130  054130  CRDMDT: .ASCIZ  /XXXXXX/
2829                                          .IFT
2830                                  CRDMG1: .ASCII  /XXXXXXXX= /
2831                                  CRDMDT: .ASCIZ  /XXXXXX/
2832                                  CRDMG2: .ASCII  <011>/  /
2833                                  CRDMD2: .ASCIZ  /XXXXXX/
2834               000                        .ENDC
2835  007416' 005015  020052  042515  CMMHDR: .ASCIZ  <015><012>/* MEM MAP */
2836               001                        .IF DF MM
2837                                          .IFF
2838         007435'                          .ODD
2839  007435'    015  043012  042522  CFMMSG: .ASCII  <015><012>/FREE MEM  /
2840                                          .IFT
2841                                          .EVEN
2842                                  CFMMSG: .ASCII  <015><012>/XXX ADRS:  FREE MEM /
2843               000                        .ENDC
2844  007451'    075  020040          CMMMSG: .ASCII  /=  /
2845               001                        .IF DF MM
2846                                          .IFF
2847  007454' 054130  054130  054130  CMMSTA: .ASCII  /XXXXXX TO /
2848  007466' 054130  054130  054130  CMMEND: .ASCII  /XXXXXX  /
2849                                          .IFT
2850                                  CMMSTA: .ASCII  /XXXXXX00 TO /
2851                                  CMMEND: .ASCII  /XXXXXX77  /
2852                                  CMMFPI: .ASCII  /X  /
2853               000                        .ENDC
2854  007476' 054130  054130  054130  CMMPNM: .ASCII  /XXXXXX  /
2855  007506' 054130  054130     000  CMMMDL: .ASCIZ  /XXXX/
2856  007513'    050  044515  046104  CMIDLM: .ASCII  /(MIDL = /
2857               001                        .IF DF MM
2858                                          .IFF
2859  007523'    130  054130  054130  CMIDL:  .ASCIZ  /XXXXXX)/
2860                                          .IFT
2861                                  CMIDL:  .ASCIZ  /XXXXXXXX)/
2862                                  CMMERM: .ASCII  <015><012>/END OF MEM = /
2863                                  CMMEMA: .ASCIZ  /XXXXXX77/
2864                                  CMMPAR: .ASCII  <015><012>/PXR'S 0 - 7 = /
2865                                  CMMPRD: .ASCII  /XXXXXX XXXXXX XXXXXX XXXXXX /
2866                                          .ASCII  /XXXXXX XXXXXX /
2867                                  TRPSP:  .ASCIZ  /XXXXXX XXXXXX/
```

```
                                                .EVEN
                                        CMMUAD: .ASCII  <015><012>/XXX ADRS:  RDIO = /
                                        CMMUA1: .ASCII  /XXXXXXXX  WRIO = /
                                        CMMUA2: .ASCIZ  /XXXXXXXX/
                                                .EVEN
                                        CMMUBM: .ASCII  <015><012>/UBM REG # /
                                        CMMUB1: .ASCII  /XX = /
                                        CMMUB2: .ASCII  /XXXXXXXX   REG # /
                                        CMMUB3: .ASCII  /XX = /
                                        CMMUB4: .ASCII  /XXXXXXXX   REG # /
                                        CMMUB5: .ASCII  /XX = /
                                        CMMUB6: .ASCII  /XXXXXXXX/
                      000               .ENDC
007533'     077 051501 047107   CADMSG: .ASCII  /?ASGN DEV: /
            007547'                     .ODD
007547'     015 050012 047522   CDSHDR: .ASCII  <015><012>/PROG # /
007560' 054130 024040           CDISPN: .ASCII  /XX (/
007564' 054130 054130 054130    CDNAME: .ASCII  /XXXXXX) DISPLAY/<015><012><015><012>
007607'     116 020117 042522   CRENPA: .ASCIZ  /NO REPT AVAIL/
007625'     077 042504 020126   CRQDRM: .ASCII  /?DEV REG = /
007640' 054130 054130 054130    CRQDRD: .ASCII  'XXXXXX / '
007651'     077 047111 020124           .ASCII  /?INT VEC = /
007664' 054130 054130 054130            .ASCII  'XXXXXX / '
007675'     077 052502 020123           .ASCII  /?BUS REQ = /
007710' 026130 020130 020057            .ASCII  'X,X / '
007716' 047101 020123 020075    CANSMG: .ASCII  /ANS = /
                001                     .IF DF MM
                                        .IFF
007724' 054130 054130 054130    CANS:   .ASCIZ  /XXXXXX/
                                        .IFT
                                CANS:   .ASCIZ  /XXXXXXXX/
                                CCDBMG: .ASCII  /ANS = /
                                CCDBAN: .ASCIZ  /XXXXXX/
                      000               .ENDC
007733'     015 042412 052116   CESMSG: .ASCIZ  <015><012>/ENTER STMNT'S/
007753'     015 050012 046507   CPCMRF: .ASCIZ  <015><012>'PGMS COMPILED / MEM REFORMATTED'
            010016'                     .EVEN
010016' 005015 052101 044440    CINTMG: .ASCII  <015><012>/AT INT: PC= /
010034' 054130 054130 054130    CINTPC: .ASCII  /XXXXXX; SP= /
010050' 054130 054130 054130    CINTSP: .ASCII  /XXXXXX; PROG= /
                001                     .IF DF MM
                                        .IFF
010066' 054130 000130           CINTPN: .ASCIZ  /XXX/
                                        .IFT
                                CINTPN: .ASCII  /XXX   /
                                CINTMD: .ASCIZ  /XXXX MODE/
                      000               .ENDC
            010073'                     .ODD
010073'     077 042122 047511   CIOSZM: .ASCII  '?RDIO = 256 / '
            010112'                     .EVEN
010112' 020077 054130 054130    CZWMSG: .ASCII  '? XXXX,X - ARE YOU SURE? (Y/N) / '
            010154'                     .EVEN
010154' 005015 042504 020126    FLLDMG: .ASCII  <015><012>'DEV ROUT "'
010170' 054130 054130 054130    FLLDNM: .ASCIZ  'XXXXXX.MPG" LOADED'
                001                     .IF DF MM
                                ADRREQ: .ASCII  '?PROG AREA ADR / '
```

```
2924                                    UMRGRQ: .ASCII  <015><012>'?UBMAP REG # (1ST OF '
2925                                    UMNREG: .ASCII  'X) / '
2926                                            .EVEN
2927                                    UMRGUD: .ASCII  <015><012>/1ST REG USED IS /
2928                                    UMRGNM: .ASCIZ  /XX/
2929                                    TRPMSG: .ASCII  <015><012>/*ER* TRAP AT VECTOR /
2930                                    TRPADR: .ASCII  /XXXXXX/<015><012><015><012>
2931                                            .ASCII  /AT TIME OF TRAP:/<015><012><015><012>/REG'S/
2932                                    TRPPSM: .ASCII  <015><012>/PSW= /
2933                                    TRPPSW: .ASCII  /XXXXXX/<015><012><015><012>/MMR0= /
2934                                    TRPMR0: .ASCII  /XXXXXX MMR2= /
2935                                    TRPMR2: .ASCII  /XXXXXX/
2936                                    TRP4OM: .ASCII  / /<015><012><015><012>/MMR1= /
2937                                    TRPMR1: .ASCII  /XXXXXX MMR3= /
2938                                    TRPMR3: .ASCIZ  /XXXXXX/
2939                                    TRPFSM: .ASCII  <015><012>/FAILING MODE:/
2940                                    MMCRLF: .ASCIZ  / /
2941                                    TRPISV: .ASCIZ  <015><012>/INT BEING SERVICED/
2942                                            .ODD
2943                                    TRPPMG: .ASCII  <015><012>/CULPRIT IS USER PROG # /
2944                                    TRPPNM: .ASCIZ  /XX/
2945                                    TRPMPG: .ASCIZ  <015><012>/CULPRIT IS MPG/
2946          000                               .ENDC
2947       010214'                              .EVEN
2948
2949                                            .LIST   BEX
```

```
                                .SBTTL  SCAN ENTERTED COMMAND DATA S/R

                        ;*****************************************************************
                        ;
                        ;  THIS S/R WILL SCAN THE DATA IN THE COMMAND FROM ITS CURRENT
                        ;  POSITION TO THE END OF THE NEXT WORD OR TO THE END OF THE LINE,
                        ;  WHICHEVER COMES FIRST. IF A WORD IS FOUND, SCANNING WILL CONTINUE
                        ;  TO DETERMINE IF THERE IS ANOTHER WORD OR IF THE ONE FOUND WAS
                        ;  THE LAST. OPTIONAL ENTRY POINTS PROVIDE FOR AUTOMATIC CONVERSION
                        ;  OF THE FOUND WORD TO EITHER PACKED DECIMAL OR BINARY (PACKED
                        ;  OCTAL). RETURN WILL BE IN-LINE UNLESS A CONVERSION IS SPECIFIED
                        ;  AND AN ERROR OCCURS ON THE CONVERSION. IN THIS CASE, RETURN
                        ;  WILL BE MADE TO THE ERROR ROUTINE.
                        ;
                        ;  LINKAGE:     JSR     R5,CSCN          SCAN FOR NEXT WORD
                        ;                         ,CSCNSP         SET DATA PNTR & SCAN ONLY
                        ;                         ,CSCNMS         SCAN & CK FOR MINUS SIGN
                        ;                         ,CSCNB8  OR
                        ;                         ,CSCNCB         SCAN & CONVERT TO BINARY
                        ;                         ,CSCNCD         SCAN & CONVERT TO PACKED DECIMAL
                        ;
                        ;  DESTROYS REGISTERS: R0,R1,R2
                        ;
                        ;  OUTPUT INFO:
                        ;
                        ;       CDSTAD  =       START ADR OF WORD IN CMND DATA
                        ;       CPNTR   =       START ADR OF NEXT WORD IN CMND OR CR/LF ADR
                        ;       CDCNT   =       NUMBER OF CHARACTERS IN WORD FOUND
                        ;       DECBUF  =       START ADR OF 3 BYTES OF DATA IN PACKED DECIMAL
                        ;       BINARY  =       ADR OF WORD WITH DATA IN BINARY
                        ;
                        ;  CSCN  &
                        ;  CSCNSP EXIT:   R0 CONTAINS ADR OF RIGHTMOST DATA BYTE IN WORD
                        ;                 R1 CONTAINS # OF CHAR'S IN WORD
                        ;                 R2 CONTAINS CSCN FLAGWORD (CSCFWD)
                        ;  CSCNCB EXIT:   R0 CONTAINS BINARY WORD
                        ;  CSCNB8 EXIT:   R0 & R2 CONTAINS 22 BIT BINARY WORD
                        ;  CSCNCD EXIT:   R0 CONTAINS ADR OF 'DECBUF' (DATA STORAGE)
                        ;
                        ;*****************************************************************

010214' 000000          CSCFWD: .WORD   0               ;SCAN S/R FLAG WORD

                        ;               = 1             ;  CONVERT TO BINARY
                        ;               = 2             ;  CONVERT TO PACKED DECIMAL
                        ;               = 4             ;  STOP SCAN ON MINUS SIGN
        000010                  CSCFND= 10              ;  DATA FOUND
        000020                  CSCTCS= 20              ;  TRAILING COMMA/SPACE
        000040                  CSCTMS= 40              ;  TRAILING MINUS SIGN
        100000                  CSCRLF= 100000          ;  CR/LF TERMINATOR
```

```
                                        ;SET UP DATA POINTER & SCAN ONLY ENTRY POINT
                                        
010216' 004767  003106          CSCNSP: JSR     PC,CSUPTR               ;INITIALIZE READ DATA POINTER
                                        
                                        ;SCAN ONLY ENTRY POINT
                                        
010222' 005002                  CSCN:   CLR     R2                      ;RESET OCTAL & DEC CONV FLAGS
010224' 000410                          BR      CSCCOM                  ;GO TO COMMON PROCESSING
010226' 012702  000004          CSCNMS: MOV     #4,R2                   ;SU FOR CK OF MINUS SIGN
010232' 000405                          BR      CSCCOM                  ;GO TO COMMON POINT
                                        
                                        ;SCAN AND CONVERT TO 16 BIT BINARY (PACKED OCTAL) ENTRY POINT
                                        
                    001                 .IF NDF MM
010234'                         CSCNB8:
                    000                 .ENDC
010234' 012702  000001          CSCNCB: MOV     #1,R2                   ;SET OCTAL FLAG & RESET DECIMAL
010240' 000402                          BR      CSCCOM                  ;GO TO COMMON PROCESSING
                                        
                                        ;SCAN AND CONVERT TO PACKED DECIMAL ENTRY POINT
                                        
010242' 012702  000002          CSCNCD: MOV     #2,R2                   ;SET DECIMAL FLAG & RESET OCTAL
010246' 005001                  CSCCOM: CLR     R1                      ;CLEAR CHARACTER COUNT
010250' 016700  171206                  MOV     CPNTR,R0                ;GET CURR POSITION ADR OF DATA
010254' 122710  000040          10$:    CMPB    #40,(R0)                ;DATA CHAR A SPACE?
010260' 001431                          BEQ     40$                     ;(N,Y-40$)
010262' 122710  000054                  CMPB    #54,(R0)                ;IS IT A COMMA?
010266' 001426                          BEQ     40$                     ;(N,Y-40$)
010270' 122710  000015                  CMPB    #15,(R0)                ;IS IT A CARR RETURN?
010274' 001442                          BEQ     60$                     ;(N,Y-60$)
010276' 122710  000012                  CMPB    #12,(R0)                ;IS IT A LINE FEED?
010302' 001437                          BEQ     60$                     ;(N,Y-60$)
010304' 122710  000055                  CMPB    #55,(R0)                ;IS IT A MINUS SIGN?
010310' 001423                          BEQ     50$                     ;N,Y-50$
010312' 032702  000010          15$:    BIT     #CSCFND,R2              ;'DATA FOUND' FLAG SET?
010316' 001004                          BNE     20$                     ;(N,Y-20$)
010320' 052702  000010                  BIS     #CSCFND,R2              ;SET THE 'FND' FLAG
010324' 010067  171246                  MOV     R0,CDSTAD               ;SAVE DATA START ADDRESS
010330' 032702  000060          20$:    BIT     #CSCTCS+CSCTMS,R2       ;'TRAILING COMMA/SPACE/- SIGN' FLAGS SET?
010334' 001024                          BNE     70$                     ;(N,Y-70$)
010336' 005201                          INC     R1                      ;ADD 1 TO CHAR COUNT
010340' 005200                  30$:    INC     R0                      ;ADD 1 TO DATA ADR
010342' 000744                          BR      10$                     ;GO CHECK NEXT CHAR
010344' 032702  000010          40$:    BIT     #CSCFND,R2              ;'FND' FLAG SET?
010350' 001773                          BEQ     30$                     ;(Y,N-30$)
010352' 052702  000020                  BIS     #CSCTCS,R2              ;SET 'TRAILING COMMA/SPACE' FLAG
010356' 000770                          BR      30$                     ;GO CHECK NEXT CHAR
010360' 032702  000004          50$:    BIT     #4,R2                   ;LOOKING FOR A MINUS SIGN?
010364' 001752                          BEQ     15$                     ;Y,N-15$
010366' 032702  000010                  BIT     #CSCFND,R2              ;"FND" FLAG SET?
010372' 001762                          BEQ     30$                     ;Y,N-30$
010374' 052702  000040                  BIS     #CSCTMS,R2              ;SET TRAILING MINUS SIGN FLG
010400' 000757                          BR      30$                     ;GO CK NEXT CHAR
010402' 052702  100000          60$:    BIS     #CSCRLF,R2              ;SET THE 'CR/LF TERMINATOR' FLAG
010406' 010267  177602          70$:    MOV     R2,CSCFWD               ;STORE FLAG WORD
010412' 010067  171044                  MOV     R0,CPNTR                ;STORE NEXT WORD/CR-LF ADR
```

```
3059  010416' 010167  171156                  MOV     R1,CDCNT                ;STORE LENGTH OF THE WORD
3060  010422' 016700  171150                  MOV     CDSTAD,R0               ;SET UP RIGHT HAND END
3061  010426' 060100                          ADD     R1,R0                   ;ADR OF DATA
3062  010430' 005300                          DEC     R0
3063  010432' 032702  000010                  BIT     #CSCFND,R2              ;WAS ANY DATA FOUND?
3064  010436' 001415                          BEQ     CSCNEX                  ;Y,N-CSCNEX
3065  010440' 032702  000003                  BIT     #3,R2                   ;EITHER CONVERT FLAG SET?
3066  010444' 001412                          BEQ     CSCNEX                  ;(Y,N-CSCNEX)
3067  010446' 004567  031676                  JSR     R5,ASDECO               ;GO CONVERT TO PACKED DECIMAL
3068  010452' 000022                          .WORD   CSCERR-.                ;CONVERSION ERROR ADR
3069  010454' 032767  000001  177532          BIT     #1,CSCFWD               ;CONVERT TO BINARY FLAG SET?
3070  010462' 001403                          BEQ     CSCNEX                  ;(Y,N-CSCVEX)
3071  010464' 004567  031442                  JSR     R5,OCTBIN               ;GO CONVERT PACKED OCTAL TO BINARY
3072  010470' 000004                          .WORD   CSCERR-.                ;CONVERSION ERROR ADR
3073  010472' 000205                  CSCNEX: RTS     R5                      ;S/R EXIT
3074                 001                      .IF DF MM
3075
3076                                          ;SCAN AND CONVERT TO 24 BIT BINARY (PACKED OCTAL) ENTRY POINT
3077
3078                                          ;INPUT & OUTPUT INFO SAME AS CSCNCB EXCEPT FOR R2
3079
3080                                  CSCNB8: JSR     R5,CSCN                 ;SCAN THE NEXT WORD
3081                                          JSR     R5,ASOCBN               ;CONVERT # TO BINARY
3082                                          .WORD   CSCERR-.
3083                                          RTS     R5                      ;EXIT IN-LINE
3084                 000                      .ENDC
3085
3086  010474' 004567  175524          CSCERR: JSR     R5,CTYPE                ;GO REPORT CONVERSION ERROR
3087  010500' 000103                          .WORD   103
3088  010502' 176105                          .WORD   CERM02-.
```

```
                                .SBTTL  VALIDATE PROGRAM # S/R

                        ;*******************************************************************
                        ;
                        ;  THE NEXT WORD IN THE CMND DATA WILL BE SCANNED AND VALIDATED
                        ;  AS A CORRECT PROGRAM NUMBER. IF VALID, THE ADDRESSES OF ITS CONTROL
                        ;  ROUTINE TABLE ENTRY AND ITS PROGRAM TABLE, IF PRESENT, WILL BE
                        ;  STORED AT STANDARD LOCATIONS. IF AN INVALID PROGRAM NUMBER, EXIT
                        ;  WILL BE MADE TO THE ERROR S/R TO REPORT THIS CONDITION.
                        ;
                        ;  LINKAGE:     JSR     R5,CVPNUM       SUBROUTINE CALL
                        ;
                        ;  DESTROYS REGISTERS: R0,R1,R2,R3,R4
                        ;
                        ;  EXIT:  CURCTE & R4 = ADDRESS OF CONTROL ROUTINE TABLE ENTRY
                        ;         CURPTA & R3 = ADDRESS OF PROGRAM TABLE
                        ;
                        ;*******************************************************************

010504' 004567 177512   CVPNUM: JSR     R5,CSCN                 ;SCAN PROG #
010510' 004567 031634            JSR     R5,ASDECO               ;CONVERT IT TO PACKED DECIMAL
010514' 000050                   .WORD   CVPER-.
010516' 004567 031374            JSR     R5,DECBIN               ;CONVERT PKED DEC TO BINARY
010522' 000042                   .WORD   CVPER-.
010524' 005700          CVPNM1:  TST     R0                      ;IS IT 0?
010526' 001416                   BEQ     CVPER                   ;N,Y-CVPER
010530' 020027 000020            CMP     R0,#MAXPRG              ;IS IT TOO LARGE?
010534' 101013                   BHI     CVPER                   ;N,Y-CVPER
010536' 010704                   MOV     PC,R4                   ;COMPUTE CNTRL ROUT TBL ENTRY
010540' 062704 023454            ADD     #CRTBL-2-.,R4
010544' 006300                   ASL     R0
010546' 060004                   ADD     R0,R4
010550' 010467 170702            MOV     R4,CURCTE               ;STORE CNTRL ROUT ENTRY ADR
010554' 011403                   MOV     (R4),R3                 ;GET PROG TBL ADR
010556' 010367 170676            MOV     R3,CURPTA               ;STORE ITS PROG TABLE ADR
010562' 000205                   RTS     R5                      ;EXIT IN-LINE

010564' 004567 175434   CVPER:   JSR     R5,CTYPE                ;GO REPORT INV PROG # ERROR
010570' 000113                   .WORD   113
010572' 176006                   .WORD   CERM01-.                ;ERROR MSG ADR
                                 .IF DF MM
                                 .PAGE
                                 .SBTTL  INITIALIZE MM PAR'S TO USER PROGRAM AREA

                        ;*******************************************************************
                        ;
                        ;  THIS SUBROUTINE USES THE CONTENTS OF R3 AS THE HIGH 16 BITS OF
                        ;  AN ABSOLUTE ADDRESS AND MAPS KERNEL'S PAR'S 4 THRU 6 TO A THREE
                        ;  PAGE AREA. UPON RETURN, R3 WILL CONTAIN THE ADDRESS OF THE FIRST
                        ;  WORD OF THIS AREA BASED UPON PAR 4. THIS ADDRESS WILL ALSO BE
                        ;  STORED AT THE LOCATION "CURPTA". THE ORIGINAL CONTENTS OF R3
                        ;  WILL ALWAYS BE STORED AT "CURPTH". IF R3 IS ZERO UPON ENTRY,
                        ;  NO OTHER ACTIONS WILL TAKE PLACE.
                        ;
                        ;  LINKAGE:     JSR     PC,UPAPAR       S/R CALL
```

```
                        ;
                        ;  DESTROYS R0,R1 - ALTERS R3
                        ;
                        ;*****************************************************************************

                        UPAPAR: MOV     R3,CURPTH               ;SAVE HI 16 BITS OF ADR
                                TST     R3                      ;R3 CONTAIN A BASE ADR?
                                BEQ     UPAPEX                  ;Y,N-UPAPEX
                                MOV     #KPAR4,R0               ;SET UP ADR OF PAR 4 REG
                                MOV     #3,R1                   ;SET UP LOOP CNT
                        10$:    MOV     R3,(R0)+                ;STORE MEM BASE ADR
                                ADD     #200,R3                 ;POINT TO NEXT MEM PAGE
                                DEC     R1                      ;DONE 3 PAGES?
                                BNE     10$                     ;Y,N-10$
                                MOV     #P4CONS,R3              ;POINT R3 AT 1ST WD OF THE AREA
                                MOV     R3,CURPTA               ;STORE IT IN MEMORY ALSO
                        UPAPEX: RTS     PC                      ;EXIT IN-LINE
                                .ENDC
```

000

```
                        .SBTTL  SHIFT HIGHER PROGRAMS UP S/R

                        ;*****************************************************************
                        ;
                        ;  THIS SUBROUNTINE WILL SEARCH THE CONTROL ROUTINE TABLE FROM
                        ;  ITS CURRENT ENTRY TO ITS END.  IF ANY HIGHER NUMBERED
                        ;  PROGRAMS ARE FOUND, THEY WILL BE RELOCATED TO THE HIGH
                        ;  END OF MEMORY OR, IF THE MEMORY MANAGEMENT VERSION, TO THE
                        ;  LOWEST FIXED ADDRESS USER PROGRAM. THE CONTROL ROUTINE TABLE
                        ;  ENTRIES WILL REFLECT THEIR NEW LOCATIONS.
                        ;
                        ;  LINKAGE:     JSR     R5,CSPGUP       SUBROUTINE CALL
                        ;
                        ;  DESTROYS REGISTERS:  R0,R1,R2
                        ;
                        ;*****************************************************************

010574' 016700  170656  CSPGUP: MOV     CURCTE,R0               ;GET CURRENT CNTRL ROUT ENTRY ADR
                  001           .IF DF MM
                                TST     CRTEXT(R0)              ;THIS A FIXED ADR PROG?
                                BPL     CSPNFP                  ;Y,N-CSPNFP
                                MOV     CENADR,R1               ;GET ITS HI 16 BITS ADR
                                MOV     R1,(R0)                 ;STORE PROG ADR IN CTRL ROUT TBL
                        CSPGSZ: MOV     R1,CURPTH               ;STORE IT
                                JSR     PC,CKF3PG               ;GET AVAIL PROG AREA SIZE IN R2
                                MOV     #6,R1                   ;CONVERT IT TO A BYTE CNT
                        5$:     ASL     R2
                                DEC     R1
                                BNE     5$
                                SUB     #2,R2                   ;LOWER ADR TO BELOW THE BNDRY
                                ADD     #P4CONS,R2              ;CONVERT IT TO A VIRTUAL ADR
                                MOV     R2,CVMEND               ;STORE IT AS VIRTUAL MEM END
                                RTS     R5                      ;EXIT IN-LINE
                  000           .ENDC
010600' 010046          CSPNFP: MOV     R0,-(SP)                ;SAVE ENTRY ADR
010602' 005046                  CLR     -(SP)                   ;RESET THE 'SHIFT' FLAG
                  001           .IF DF MM
                                MOV     CPAREA,CSPNPA           ;INIT NEXT PROG AREA ADR
                                MOV     CRMEND,CSPLFP           ;INIT LOWEST FIXED PROG ADR
                                MOV     PC,R0                   ;SET UP CNTRL ROUT TBL START ADR
                                ADD     #CRTBL-2-.,R0           ;BEGIN SCAN AT BEGINNING OF TBL
                  000           .ENDC
010604' 005720          10$:    TST     (R0)+                   ;INCR TO NXT TBL ENTRY
010606' 022710  177777          CMP     #177777,(R0)            ;END OF THE TABLE?
010612' 001404                  BEQ     20$                     ;N,Y-20$
010614' 005710                  TST     (R0)                    ;ENTRY IN USE?
010616' 001772                  BEQ     10$                     ;Y,N-10$
                  001           .IF DF MM
                                .IFT
                                TST     CRTEXT(R0)              ;THIS A FIXED ADR PROG?
                                BMI     15$                     ;N,Y-15$
                                CMP     R0,2(SP)                ;BELOW ORG PROG SLOT?
                                BLO     12$                     ;N,Y-12$
                                BEQ     10$                     ;AT ORG SLOT? (N,Y-10$)
                                MOV     R0,(SP)                 ;SET SHIFT FLAG
```

```
3222                                            BR      10$             ;GO CK NXT ENTRY
3223                                    12$:    MOV     (R0),CSPNPA     ;CALC & SAVE MEM ADR OF AREA
3224                                            ADD     CRTEXT(R0),CSPNPA ;AFTER THIS PROG
3225                                            BR      10$             ;GO CK NXT SLOT
3226                                            .IFF
3227  010620' 010016                            MOV     R0,(SP)         ;SET SHIFT FLAG
3228  010622' 000770                            BR      10$             ;GO CK NXT ENTRY
3229                                            .IFT
3230                                    15$:    CMP     (R0),CSPLFP     ;THIS PROG THE LOWEST SO FAR?
3231                                            BHIS    10$             ;Y,N-10$
3232                                            MOV     (R0),CSPLFP     ;SAVE ITS ADR AND
3233                                            DEC     CSPLFP          ;POINT IT TO NXT LOW 32 WD BLK
3234                                            BR      10$             ;GO CK NEXT ENTRY
3235                    000                     .ENDC
3236  010624' 005726                    20$:    TST     (SP)+           ;'SHIFT' FLAG SET?
3237  010626' 001006                            BNE     25$             ;N,Y-25$
3238  010630' 012601                            MOV     (SP)+,R1        ;GET ORG TBL ENTRY ADR
3239  010632' 005711                            TST     (R1)            ;PROG ALREADY EXIST FOR ORG ENTRY?
3240  010634' 001022                            BNE     40$             ;N,Y-40$
3241                    001                     .IF DF MM
3242                                            .IFF
3243  010636' 016711  170604                    MOV     CFMST,(R1)      ;SET UP START ADR OF FREE MEM
3244                                            .IFT
3245                                            MOV     CSPNPA,(R1)     ;GET ADR OF NXT MEM AREA
3246                    000                     .ENDC
3247  010642' 000417                            BR      40$             ;GO TO S/R EXIT
3248  010644' 012601                    25$:    MOV     (SP)+,R1        ;GET ORG TBL ENTRY ADR
3249  010646' 010346                            MOV     R3,-(SP)        ;SAVE WORK REG'S
3250  010650' 010446                            MOV     R4,-(SP)
3251                    001                     .IF DF MM
3252                                            .IFF
3253  010652' 016704  170572                    MOV     CFMEND,R4       ;GET END OF FREE MEM ADR
3254                                            .IFT
3255                                            MOV     CSPLFP,R4       ;GET ADR OF LOWEST FIXED PROG
3256                    000                     .ENDC
3257  010656' 005740                    30$:    TST     -(R0)           ;DECR TBL ENTRY PNTR TO NXT LOWEST
3258  010660' 020001                            CMP     R0,R1           ;BACK TO ORG ENTRY?
3259  010662' 001012                            BNE     50$             ;Y,N-50$
3260                    001                     .IF NDF MM
3261  010664' 010467  170560                    MOV     R4,CFMEND       ;STORE NEW FREE MEM END ADR
3262                    000                     .ENDC
3263  010670' 005711                            TST     (R1)            ;PROG ALREADY EXIST FOR ORG ENTRY?
3264  010672' 001001                            BNE     35$             ;N,Y-35$
3265  010674' 010311                            MOV     R3,(R1)         ;SET UP START ADR FOR THIS PROG
3266  010676' 012604                    35$:    MOV     (SP)+,R4        ;RESTORE WORK REG'S
3267  010700' 012603                            MOV     (SP)+,R3
3268                    001                     .IF DF MM
3269                                            .IFF
3270  010702' 011167  170552            40$:    MOV     (R1),CURPTA     ;STORE NEW CURR PROG TBL ADR
3271  010706' 000205                            RTS     R5              ;EXIT IN-LINE
3272                                            .IFT
3273                                    40$:    MOV     (R1),R1         ;GET PROG AREA ADR IN R1
3274                                            BR      CSPGSZ          ;GO COMPUTE AREA SIZE & EXIT
3275                    000                     .ENDC
3276  010710' 005710                    50$:    TST     (R0)            ;IS ENTRY IN USE?
3277  010712' 001761                            BEQ     30$             ;Y,N-30$
```

```
                    001
                    000
010714' 011003
                    001
010716' 016302  000026                  .IF DF MM
010722' 060203                          TST     CRTEXT(R0)          ;FIXED ADR PROG?
010724' 006202                          BMI     30$                 ;N,Y-30$
010726' 005724                          .ENDC
010730' 014344                  60$:    MOV     (R0),R3             ;GET PROG'S START ADR
010732' 005302                          .IF DF MM
010734' 001375                          .IFF
010736' 010402                          MOV     PLNGTH(R3),R2       ;GET PROG'S LENGTH
010740' 160302                          ADD     R2,R3               ;SET UP PROG END ADR
010742' 060264  000016                  ASR     R2                  ;DIVIDE LNGTH BY 2
010746' 060264  000020                  TST     (R4)+               ;ADD 2 TO END OF FREE MEM ADR
010752' 060264  000022                  MOV     -(R3),-(R4)         ;MOVE WORD OF THE PROG
010756' 060264  000024                  DEC     R2                  ;DECR WORD CNT BY 1
010762' 010410                          BNE     60$                 ;CNT = 0? (Y,N-60$)
010764' 005744                          MOV     R4,R2               ;COMPUTE RELOCATION FACTOR
010766' 000733                          SUB     R3,R2
                                        ADD     R2,PRDIOA(R4)       ;ADJ ADR'S OF SHIFTED PROG
                                        ADD     R2,PWRIOA(R4)
                                        ADD     R2,PSRCST(R4)
                                        ADD     R2,POBJST(R4)
                                        MOV     R4,(R0)             ;SET UP NEW PROG ADR IN CTRL ROUT TBL
                                        TST     -(R4)               ;SET UP NEW END OF FREE MEM ADR
                                        BR      30$                 ;GO CK NXT TBL ENTRY
                                        .IFT
                                        MOV     CRTEXT(R0),R2       ;GET PROG'S LENGTH
                                        MOV     R0,-(SP)            ;SAVE R0 & R1
                                        MOV     R1,-(SP)
                                        JSR     PC,MOVPGU           ;MOVE THE PROG UP IN MEM
                                        MOV     (SP)+,R1            ;RESTORE R0 & R1
                                        MOV     (SP)+,R0
                                        MOV     R4,(R0)             ;STORE ITS NEW ADR
                                        DEC     R4                  ;POINT TO NXT LOW 32 WD BLK
                                        BR      30$                 ;GO CK NXT TBL ENTRY


                                CSPNPA: .WORD   0                   ;*;NEXT PROG AREA ADR
                                CSPLFP: .WORD   0                   ;*;ADR OF LOWEST FIXED ADR PROG
                    000                 .ENDC
```

```
                                .SBTTL  SHIFT PROGRAMS DOWN & RECOMPILE S/R

                        ;*****************************************************************
                        ;
                        ;  THIS S/R PERFORMS THREE BASIC FUNCTIONS.  FIRST, IT
                        ;  WILL SCAN THE CONTROL ROUTINE TABLE FOR EXISTING PROGRAMS AND
                        ;  DETERMINES IF THERE ARE ANY UNUSED MEMORY AREAS.  IF THERE
                        ;  ARE, PROGRAMS WILL BE SHIFTED DOWN TO ELIMINATE THESE AREAS.
                        ;  AFTER THE PROGRAMS HAVE BEEN SHIFTED DOWN, IF ANY, THE
                        ;  FLAGWORD FOR EACH PROGRAM WILL BE RE-INITIALIZED SO THAT
                        ;  PROGRAMS CAN ONLY BE INITIATED BY THE "RUN" COMMAND AND
                        ;  ALL PROGRAMS WILL THEN BE RECOMPILED.
                        ;
                        ;  LINKAGE:     JSR     R5,CSDNRC       SUBROUTINE CALL
                        ;
                        ;  DESTROYS REGISTERS: R0,R1,R2
                        ;
                        ;*****************************************************************

010770' 010346  CSDNRC: MOV     R3,-(SP)                ;SAVE WORK REG'S
010772' 010446          MOV     R4,-(SP)
010774' 010546          MOV     R5,-(SP)
010776' 005046          CLR     -(SP)                   ;RESET 'END SHIFT' FLAG
011000' 016705  170450  MOV     CPAREA,R5               ;INITIALIZE PREV PROG END ADR
                                                        ;TO START OF PROG AREA
            001         .IF DF MM
                        MOV     R5,(PC)+                ;INIT HIGHEST MPG PROG ADR
                HIMPGP: .WORD   0
            000         .ENDC
011004' 010704          10$:    MOV     PC,R4           ;SET UP START ADR OF CNTRL
011006' 062704  023210          ADD     #CRTBL-.,R4     ;ROUT TBL
011012' 005714          20$:    TST     (R4)            ;THIS ENTRY IN USE?
011014' 001021          BNE     50$                     ;N,Y-50$
            001         .IF DF MM
                        CLR     CRTEXT(R4)              ;HSKP ITS TBL EXTENSION WORD
            000         .ENDC
011016' 005724          30$:    TST     (R4)+           ;INCR TO NXT TBL ENTRY
011020' 022714  177777          CMP     #177777,(R4)    ;END OF CNTRL ROUT TBL?
011024' 001372          BNE     20$                     ;Y,N-20$
011026' 005726          TST     (SP)+                   ;'END SHIFT' FLAG SET?
011030' 001007          BNE     40$                     ;N,Y-40$
011032' 010446          MOV     R4,-(SP)                ;SET END SHIFT FLAG
            001         .IF DF MM
                        .IFF
011034' 016767  170412  170406  MOV     CRMEND,CFMEND   ;SET FREE MEM END TO REAL MEM END
011042' 010567  170400          MOV     R5,CFMST        ;MOVE PREV PROG END TO FREE MEM START
                        .IFT
                        JSR     PC,FINDFM               ;GO DETERMINE FREE MEM AREA
                        MOV     R0,CFMST                ;STORE ITS START ADR
                        ADD     R0,R1
                        DEC     R1
                        MOV     R1,CFMEND               ;STORE ITS END ADR
                        JSR     PC,SUMIFM               ;SET UP MIDL & FREE'S ADRS
            000         .ENDC
011046' 000756          BR      10$                     ;GO DO SECOND PASS OF TBL
```

```
3373  011050' 012605                  40$:    MOV     (SP)+,R5                ;RESTORE WORK REG'S
3374  011052' 012604                          MOV     (SP)+,R4
3375  011054' 012603                          MOV     (SP)+,R3
3376  011056' 000205                          RTS     R5                      ;EXIT IN-LINE
3377  011060' 011403                  50$:    MOV     (R4),R3                 ;GET THIS PROG TBL ADR
3378              001                          .IF DF MM
3379                                           JSR     PC,UPAPAR               ;MAP KERNEL'S PAR'S TO USR PROG
3380              000                          .ENDC
3381  011062' 005716                          TST     (SP)                    ;'END SHIFT' FLAG SET?
3382  011064' 001411                          BEQ     60$                     ;Y,N-60$
3383  011066' 011463  000004                  MOV     (R4),PFWADR(R3)         ;STORE FLAGWORD ADR IN PROG TBL
3384              001                          .IF DF MM
3385                                           JSR     PC,MAPUPG               ;MAP USER'S PAR & PDR STORAGE
3386                                           MOV     (R4),(PC)+              ;GET HIGH 16 BITS ADR
3387                                  52$:     .WORD   XXXX
3388                                           JSR     R5,G22BAD               ;CONVERT IT TO A 22 BIT ADR
3389                                           .WORD   52$-.
3390                                           MOV     R2,PGABAD               ;STORE PROG'S ABS ADR
3391                                           MOV     R0,PGABAD+2             ;FOR COMPILER USE
3392                                           MOV     R3,R1                   ;POINT AT STORAGE AREA FOR
3393                                           ADD     #PRDIOX,R1              ;18/22 BIT ABS ADR OF RDIO
3394                                           MOV     R2,(R1)+                ;STORE HIGH BITS OF BASE ADR
3395                                           MOV     PRDIOA(R3),(R1)         ;STORE RDIO DISPL
3396                                           ADD     R0,(R1)                 ;ADD IN LOW BITS OF BASE ADR
3397                                           ADC     -(R1)
3398                                           ADD     #8.,R1                  ;POINT AT WRIO STORAGE
3399                                           MOV     R2,(R1)+                ;STORE HIGH BITS OF BASE ADR
3400                                           MOV     PWRIOA(R3),(R1)         ;STORE WRIO DISPL
3401                                           ADD     R0,(R1)                 ;ADD IN LOW BITS OF BASE ADR
3402                                           ADC     -(R1)
3403                                           BIT     #UNIMAP,CSYSFW          ;WE USING THE UNIBUS MAP?
3404                                           BEQ     58$                     ;Y,N-58$
3405                                           TST     PUBMAP(R3)              ;DOES PROG USE UNIBUS MAP?
3406                                           BEQ     58$                     ;Y,N-58$
3407                                           MOVB    PUBMAP(R3),R1           ;GET 1ST MAP REG #
3408                                           ASL     R1                      ;COMPUTE DISPL INTO UNIBUS
3409                                           ASL     R1                      ;MAP REGS
3410                                           ADD     #UBMAP0,R1
3411                                           MOVB    PUBMAP+1(R3),-(SP)      ;GET # OF MAP REG BITS
3412                                           CMPB    (SP),#3                 ;HOW MANY REGS BEING USED?
3413                                           BLO     54$                     ;BR IF 1
3414                                           BEQ     53$                     ;BR IF 2
3415                                           SUB     #3,(SP)                 ;ADJ FOR 3 REGS
3416                                  53$:     DECB    (SP)                    ;ADJ FOR 2 REGS
3417                                  54$:     MOV     R0,(R1)+                ;STORE UNIBUS MAP REG VALUE
3418                                           MOV     R2,(R1)+
3419                                           ADD     #20000,R0               ;POINT TO NEXT 4K WORDS
3420                                           ADC     R2
3421                                           DECB    (SP)                    ;DONE ALL REGS?
3422                                           BNE     54$                     ;Y,N-54$
3423                                           TST     (SP)+                   ;HSKP STK
3424                                           MOVB    PUBMAP(R3),R2           ;GET 1ST UNIBUS MAP REG #
3425                                           CLR     R0
3426                                           MOV     #3,R1                   ;SET UP LOOP COUNT
3427                                  56$:     ASR     R2                      ;CONVERT REG # TO UNIBUS MAP
3428                                           ROR     R0                      ;REG SELECT BITS
```

```
                                        DEC     R1
                                        BNE     56$
                                        MOV     R2,PGUBMA               ;STORE UNIMAP REG SELECT BITS FOR
                                        MOV     R0,PGUBMA+2             ;COMPILER USE
                                        MOV     R3,R1                   ;POINT AT STORAGE AREA FOR 18
                                        ADD     #PRDIOV,R1              ;BIT VIRT ADRS OF RDIO & WRIO
                                        MOV     R2,(R1)+                ;STORE HIGH REG SELECT BITS
                                        MOV     PRDIOA(R3),(R1)         ;STORE RDIO DISPL
                                        ADD     R0,(R1)                 ;ADD IN LOW REG SELECT BITS
                                        ADC     -(R1)
                                        ADD     #8.,R1                  ;POINT AT WRIO STORAGE
                                        MOV     R2,(R1)+                ;STORE HIGH REG SELECT BITS
                                        MOV     PWRIOA(R3),(R1)         ;STORE WRIO DISPL
                                        ADD     R0,(R1)                 ;ADD IN LOW REG SELECT BITS
                                        ADC     -(R1)
                    000                 .ENDC
011072' 032713  000100          58$:    BIT     #OCPRES,(R3)            ;THIS PROG ALREADY HAVE OBJ CODE?
011076' 001747                          BEQ     30$                     ;Y,N-30$
                    001                 .IF DF MM
                                        .IFF
011100' 005013                          CLR     (R3)                    ;RESET THIS PROG'S FLGWD
                                        .IFT
                                        BIC     #177777-USEUBM,(R3)     ;HSKP THIS PROG'S FLGWD
                    000                 .ENDC
011102' 004567  003722                  JSR     R5,COMPIL               ;GO RECOMPILE THIS PROG
011106' 000743                          BR      30$                     ;GO CK NXT TBL ENTRY
011110' 016301  000026          60$:    MOV     PLNGTH(R3),R1           ;GET PROG'S LENGTH
011114' 001002                          BNE     65$                     ;LENGTH = 0? (Y,N-65$)
011116' 005014                          CLR     (R4)                    ;RESET ITS CTRL ROUT TBL ENTRY
                    001                 .IF DF MM
                                        .IFF
011120' 000736                          BR      30$                     ;GO CK NXT PROG
011122' 020305                  65$:    CMP     R3,R5                   ;PROG'S START ADR = PREV PROG END ADR?
011124' 001002                          BNE     70$                     ;Y,N-70$
011126' 060105                          ADD     R1,R5                   ;ADD PROG LENGTH TO ITS START ADR &
                                                                        ;SAVE AS PREV PROG END ADR
011130' 000732                          BR      30$                     ;GO CHECK NEXT PROG SLOT
011132' 010302                  70$:    MOV     R3,R2                   ;COMPUTE RELOCATION FACTOR
011134' 160502                          SUB     R5,R2
011136' 160263  000016                  SUB     R2,PRDIOA(R3)           ;ADJ PROG TBL ENTRIES TO NEW LOC
011142' 160263  000020                  SUB     R2,PWRIOA(R3)
011146' 160263  000022                  SUB     R2,PSRCST(R3)
011152' 160263  000024                  SUB     R2,POBJST(R3)
011156' 006001                          ROR     R1                      ;DIVIDE PROG LENGTH BY 2
011160' 010514                          MOV     R5,(R4)                 ;SET NEW PROG ADR IN CNTRL ROUT TBL
011162' 012325                  80$:    MOV     (R3)+,(R5)+             ;MOV 1 WORD OF PROG DOWN
011164' 005301                          DEC     R1                      ;DECR WORD COUNT
011166' 001375                          BNE     80$                     ;CNT = 0? (Y,N-80$)
011170' 000712                          BR      30$                     ;GO CHECK NXT PROG SLOT
                                        .IFT
                                        CLR     CRTEXT(R4)              ;RESET CNTRL ROUT TBL EXT
                                        BR      80$                     ;GO CK NXT PROG
                                65$:    TST     CRTEXT(R4)              ;THIS A FIXED ADR PROG?
                                        BMI     80$                     ;N,Y-80$
                                        CMP     (R4),R5                 ;PROG'S START ADR = PREV PROG END ADR?
                                        BNE     70$                     ;Y,N-70$
```

```
3485                                    ADD     CRTEXT(R4),R5           ;ADD PROG LGTH TO ITS START ADR &
3486                                                                    ;SAVE AS PREV PROG ADR
3487                                    BR      75$                     ;GO STORE NEW ADR
3488                            70$:    MOV     CRTEXT(R4),R3           ;GET PROG'S LENGTH
3489                                    JSR     PC,MOVPGD               ;MOVE THE PROG DOWN
3490                            75$:    MOV     R5,HIMPGP               ;STORE AS HIGHEST MPG PROG ADR
3491                            80$:    JMP     30$                     ;GO CK NXT PROG SLOT
3492
3493                            PGABAD: .WORD   0                       ;UP TO 22 BIT ABSOLUTE ADR (16 - 21)
3494                                    .WORD   0                       ;OF CURRENT PROG'S AREA (0 - 15)
3495                            PGUBMA: .WORD   0                       ;CURRENT PROG'S UNIBUS MAP (16 & 17)
3496                                    .WORD   0                       ;REG SELECT BITS (0 - 15)
3497           000                      .ENDC
```

```
                                .SBTTL  REQUEST DEVICE & INT ADR'S S/R

                        ;*****************************************************************
                        ;
                        ;  THIS S/R USES A 4 WORD AREA AS INPUT. THESE WORDS SHOULD
                        ;  CONTAIN THE DEV REG ADR, INT VECTOR ADR, SINGLE OR READ
                        ;  INT PROCESSOR STATUS WORD, AND THE WRITE INT STATUS WORD
                        ;  (OR A WORD OF ZEROS) IN THAT ORDER. REG R4 MUST CONTAIN AN
                        ;  ADDRESS THAT POINTS TO THE FIRST WORD.
                        ;
                        ;  THE THREE QUERY MESSAGES WILL BE DISPLAYED WITH THE
                        ;  EXISTING DATA BEING TAILORED AND INCLUDED. THE USER
                        ;  MAY MAKE AN ENTRY THAT REPLACES THE EXISTING DATA OR MAKE A
                        ;  NULL ENTRY WHICH CAUSES THE EXISTING DATA TO BE RETAINED.
                        ;
                        ;  LINKAGE:     JSR     R5,CRQADR       SUBROUTINE CALL
                        ;
                        ;  INPUT:       R4 MUST CONTAIN ADR OF DEVICE'S 4 INFO WORDS
                        ;
                        ;  DESTROYS REGISTERS: R0,R1,R2
                        ;
                        ;*****************************************************************

011172' 122777  000012  170262  CRQADR: CMPB    #012,@CPNTR             ;LINE FEED TERMINATE PREV REPLY?
011200' 001002                          BNE     2$                      ;Y,N-2$
011202' 000167  000400                  JMP     CRQEX                   ;GO TO EXIT - BYPASS REQUESTS
011206' 010346                  2$:     MOV     R3,-(SP)                ;SAVE WORK REG'S
011210' 010446                          MOV     R4,-(SP)
011212' 010546                          MOV     R5,-(SP)
011214' 005046                          CLR     -(SP)                   ;RESET LOOP CNT
011216' 012703  176265          5$:     MOV     #CRQDRM-CRQLNK,R3       ;SET UP ADR OF FIRST MSG
011222' 010705                          MOV     PC,R5                   ;SET UP ADR OF DATA IN 1ST MSG
011224' 062705  176414                  ADD     #CRQDRD-.,R5
011230' 010702                  CRQNXM: MOV     PC,R2                   ;SET UP LINK INFO ADR
011232' 062702  000106                  ADD     #CRQLNK-.,R2
011236' 010322                          MOV     R3,(R2)+                ;STORE MSG ADR IN LINK INFO
011240' 012712  000013                  MOV     #11.,(R2)               ;SET UP MSG ONLY LENGTH
011244' 011400                          MOV     (R4),R0                 ;GET PRESET DATA WORD
011246' 122716  000002                  CMPB    #2,(SP)                 ;THIS BUS REQ MSG?
011252' 001406                          BEQ     10$                     ;N,Y-10$
011254' 062712  000011                  ADD     #9.,(R2)                ;SET UP MSG & DATA LENGTH
011260' 010501                          MOV     R5,R1                   ;SET UP OUTPUT ADR FOR S/R
011262' 004567  027640                  JSR     R5,BINAS1               ;GO CONVERT DATA TO ASCII
011266' 000421                          BR      CRQLPT                  ;GO ISSUE DEV REG OR INT VECT MSG
011270' 062712  000006          10$:    ADD     #6.,(R2)                ;SET UP BUS REQ MSG LNGTH
011274' 004367  000310                  JSR     R3,CRQCTA               ;CONVERT FIRST PS WD TO ASCII #
011300' 016400  000002                  MOV     2(R4),R0                ;GET WRITE PS WORD
011304' 005700                          TST     R0                      ;ANY DATA?
011306' 001005                          BNE     20$                     ;N,Y-20$
011310' 012701  000040                  MOV     #40,R1                  ;CLEAR 2ND BR TO SPACES
011314' 110125                          MOVB    R1,(R5)+
011316' 110115                          MOVB    R1,(R5)
011320' 000404                          BR      CRQLPT                  ;GO ISSUE SINGLE BR MSG
011322' 112725  000054          20$:    MOVB    #054,(R5)+              ;MOVE COMMA TO MSG
011326' 004367  000256                  JSR     R3,CRQCTA               ;CONVERT SECOND PS WD TO ASCII
```

```
3555  011332'  004567  174666          CRQLPT:  JSR     R5,CTYPE            ;ISSUE INFO REQUEST MSG
3556  011336'  000420                           .WORD   420
3557  011340'  000000                   CRQLNK: .WORD   XXXX
3558  011342'  000000                           .WORD   XXXX
3559  011344'  004577  166460                   JSR     R5,@CTRD            ;ISSUE READ FOR THE REPLY
3560  011350'  170114                           .WORD   CCMDRD-.
3561  011352'  000107                           .WORD   107
3562  011354'  004767  001750                   JSR     PC,CSUPTR           ;INITIALIZE READ DATA PNTR
3563  011360'  004567  176636          25$:     JSR     R5,CSCN             ;SCAN FOR REPLY
3564  011364'  032702  000010                   BIT     #CSCFND,R2          ;ANY DATA ENTERED ON REPLY?
3565  011370'  001465                           BEQ     60$                 ;Y,N-60$
3566  011372'  004567  030752          30$:     JSR     R5,ASDECO           ;CONVERT DATA TO BINARY
3567  011376'  000232                           .WORD   CRQER-.
3568  011400'  004567  030526                   JSR     R5,OCTBIN
3569  011404'  000224                           .WORD   CRQER-.
3570  011406'  122716  000002                   CMPB    #2,(SP)             ;THIS BUS REQ MSG?
3571  011412'  001026                           BNE     44$                 ;Y,N-44$
3572  011414'  020027  000007                   CMP     R0,#7               ;VALID BUSS REQ #?
3573  011420'  101405                           BLOS    40$                 ;N,Y-40$
3574  011422'  004567  174576                   JSR     R5,CTYPE            ;ISSUE 'INV BR' MSG
3575  011426'  000712                           .WORD   712
3576  011430'  175416                           .WORD   CERM20-.
3577  011432'  000737                           BR      CRQLPT              ;GO ISSUE MSG AGAIN
3578  011434'  000300                   40$:    SWAB    R0                  ;SET UP PS WORD
3579  011436'  006200                           ASR     R0
3580  011440'  006200                           ASR     R0
3581  011442'  006200                           ASR     R0
3582  011444'  010024                           MOV     R0,(R4)+            ;STORE NEW PS WORD
3583  011446'  032716  000400                   BIT     #400,(SP)           ;WAS THIS 2ND PS WORD?
3584  011452'  001035                           BNE     62$                 ;N,Y-62$
3585  011454'  005767  176534                   TST     CSCFWD              ;IS THERE ANOTHER BR # ENTERED?
3586  011460'  100432                           BMI     62$                 ;Y,N-62$
3587  011462'  052716  000400                   BIS     #400,(SP)           ;SET SECOND PS WORD FLAG
3588  011466'  000734                           BR      25$                 ;GO PROCESS SECOND PS WORD
3589  011470'  032700  000001          44$:     BIT     #1,R0               ;IS MEM ADR ODD?
3590  011474'  001405                           BEQ     48$                 ;Y,N-48$
3591  011476'  004567  174522                   JSR     R5,CTYPE            ;ISSUE 'ODD ADR' ERR MSG
3592  011502'  000702                           .WORD   702
3593  011504'  175435                           .WORD   CERM25-.
3594  011506'  000711                           BR      CRQLPT              ;GO RE-ISSUE MSG
3595  011510'  105716                   48$:    TSTB    (SP)                ;THIS DEV REG ADR MSG?
3596  011512'  001010                           BNE     54$                 ;Y,N-54$
3597  011514'  020027  160010                   CMP     R0,#160010          ;IS ADR IN DEV REG RANGE?
3598  011520'  103010                           BHIS    58$                 ;N,Y-58$
3599  011522'  004567  174476          50$:     JSR     R5,CTYPE            ;ISSUE 'INV ADR RANGE' MSG
3600  011526'  000712                           .WORD   712
3601  011530'  175421                           .WORD   CERM26-.
3602  011532'  000677                           BR      CRQLPT              ;GO RE-ISSUE MSG
3603  011534'  020027  001000          54$:     CMP     R0,#1000            ;IS ADR IN INT VECT RANGE?
3604  011540'  103370                           BHIS    50$                 ;Y,N-50$
3605  011542'  010014                   58$:    MOV     R0,(R4)             ;STORE ENTERED DATA
3606  011544'  005724                   60$:    TST     (R4)+               ;INCR TO NXT STORAGE ADR
3607  011546'  005216                   62$:    INC     (SP)                ;ADD 1 TO LOOP COUNT
3608  011550'  062703  000024                   ADD     #20.,R3             ;SET MSG PNTR TO NXT MSG
3609  011554'  062705  000024                   ADD     #20.,R5             ;SET TAILORED DATA PNTR TO NXT MSG
3610  011560'  122777  000012  167674          CMPB    #012,@CPNTR         ;LINE FEED TERMINATE PREV REPLY?
```

```
3611  011566' 001403                    BEQ     70$                     ;N,Y-70$
3612  011570' 122716  000003            CMPB    #3,(SP)                 ;DONE ALL MSG'S?
3613  011574' 001215                    BNE     CRQNXM                  ;Y,N-CRQNXM
3614  011576' 005726            70$:    TST     (SP)+                   ;TAKE FLGWD OFF THE STACK
3615  011600' 012605                    MOV     (SP)+,R5                ;RESTORE REG'S
3616  011602' 012604                    MOV     (SP)+,R4
3617  011604' 012603                    MOV     (SP)+,R3
3618  011606' 000205            CRQEX:  RTS     R5                      ;EXIT IN-LINE
3619
3620
3621  011610' 006300            CRQCTA: ASL     R0                      ;CONVERT PRIORITY FIELD TO
3622  011612' 006300                    ASL     R0                      ;TO A 1 DIGIT ASCII NUMBER
3623  011614' 006300                    ASL     R0
3624  011616' 000300                    SWAB    R0
3625  011620' 052700  000060            BIS     #60,R0
3626  011624' 110025                    MOVB    R0,(R5)+                ;STORE BUS REQ # IN MSG
3627  011626' 000203                    RTS     R3                      ;EXIT IN-LINE
3628
3629
3630  011630' 004567  174370    CRQER:  JSR     R5,CTYPE                ;ISSUE DATA CONV ERR MSG
3631  011634' 000702                    .WORD   702
3632  011636' 174751                    .WORD   CERM02-.
3633  011640' 000634                    BR      CRQLPT                  ;GO ISSUE MSG AGAIN
```

```
                                .SBTTL  LOAD DEVICE ROUTINE S/R

                        ;*****************************************************************
                        ;
                        ;  THIS S/R WILL LOAD THE SPECIFIED DEVICE ROUTINE FROM THE
                        ;  SYSTEM LOAD DEVICE AND PLACE IT IN MEMORY IMMEDIATELY FOLLOWING
                        ;  THE CURRENT PROGRAM TABLE. INITIALIZATION OF CERTAIN PROGRAM
                        ;  TABLE ENTRIES WILL ALSO OCCUR. THE 'REQUEST ADR'S' S/R WILL
                        ;  BE USED TO UPDATE THE DEV ROUTINE'S DEVICE ADR'S.
                        ;
                        ;  LINKAGE:     JSR     R5,CLDEVR       SUBROUTINE CALL
                        ;               .WORD   ERRADR-.        INSUF MEM REL ERR RET ADR
                        ;               .WORD   DERADR-.        DEVICE ERROR REL ERR RET ADR
                        ;
                        ;  INPUT:       R3 MUST CONTAIN ADR OF CURRENT PROG TABLE
                        ;
                        ;  DESTROYS REGISTERS:  R0,R1,R2
                        ;
                        ;*****************************************************************

011642' 012767  000400  000544  CLDEVR: MOV     #CIOALT,RDIOSZ          ;INITIALIZE I/O AREA SIZES
011650' 012767  000400  000546          MOV     #CIOALT,WRIOSZ
011656' 122777  000012  167576          CMPB    #012,@CPNTR             ;BYPASS PRESET MESSAGES?
011664' 001460                          BEQ     CLDPDR                  ;N,Y-CLDPDR
011666' 012767  042122  176200          MOV     #"RD,CIOSZM+1           ;INITIALIZE MSG TO "RDIO"
011674' 004567  174324          CSZREQ: JSR     R5,CTYPE                ;ISSUE REQ MSG
011700' 000420                          .WORD   420
011702' 176171                          .WORD   CIOSZM-.
011704' 000016                          .WORD   14.
011706' 004577  166116                  JSR     R5,@CTRD                ;ISSUE RD FOR THE REPLY
011712' 167552                          .WORD   CCMDRD-.
011714' 000036                          .WORD   30.
011716' 004567  176274                  JSR     R5,CSCNSP               ;SCAN FOR THE REPLY
011722' 032702  000010                  BIT     #CSCFND,R2              ;DATA RECEIVED?
011726' 001433                          BEQ     115$                    ;Y,N-115$
011730' 004567  030414                  JSR     R5,ASDECO               ;CONVERT DECIMAL DATA TO BINARY
011734' 000552                          .WORD   CLDER3-.
011736' 004567  030154                  JSR     R5,DECBIN
011742' 000544                          .WORD   CLDER3-.
011744' 032700  000001                  BIT     #1,R0                   ;WAS AN ODD SIZE ENTERED?
011750' 001401                          BEQ     100$                    ;Y,N-100$
011752' 005200                          INC     R0                      ;MAKE IT EVEN
011754' 122767  000122  176112  100$:   CMPB    #'R,CIOSZM+1            ;THIS RDIO MSG?
011762' 001012                          BNE     110$                    ;Y,N-110$
011764' 010067  000424                  MOV     R0,RDIOSZ               ;SAVE DATA AS RDIO SIZE
011770' 122777  000012  167464  105$:   CMPB    #012,@CPNTR             ;LINE FEED TERMINATE RDIO REPLY?
011776' 001413                          BEQ     CLDPDR                  ;N,Y-CLDPDR
012000' 012767  051127  176066          MOV     #"WR,CIOSZM+1           ;SET UP FOR WRIO MSG
012006' 000732                          BR      CSZREQ                  ;GO GET WRIO SIZE
012010' 010067  000410          110$:   MOV     R0,WRIOSZ               ;STORE DATA AS WRIO SIZE
012014' 000404                          BR      CLDPDR                  ;GO TO DVR LOAD
012016' 122767  000122  176050  115$:   CMPB    #'R,CIOSZM+1            ;THIS RDIO MSG?
012024' 001761                          BEQ     105$                    ;N,Y-105$
                    001                 .IF DF MM
                                        .IFT
```

```
                                        CLDPDR: MOV     RDIOSZ,R0               ;COMBINE SIZES OF RDIO AND
                                                ADD     WRIOSZ,R0               ;WRIO
                                                CMP     R0,#24250.              ;THEIR SIZE GREATER THAN 2.9 PAGES?
                                                BLOS    CLDPDX                  ;Y,N-CLDPDX
                                                JSR     R5,CTYPE                ;ISSUE ERROR MSG
                                                .WORD   402
                                                .WORD   CERM33-.
                                                CLRB    @CPNTR                  ;RESET LINE FEED TERM IF THERE
                                                BR      CLDEVR                  ;GO REQUEST NEW VALUES
                                        CLDPDX: MOV     R3,R2                   ;GET DEV ROUT AREA START ADR
                                                .IFF
012026' 010302                          CLDPDR: MOV     R3,R2                   ;GET DEV ROUT AREA START ADR
                000                             .ENDC
012030' 062702  000242                          ADD     #PTLGTH,R2
012034' 010267  000202                          MOV     R2,CDVRST               ;STORE IT IN S/R LINK INFO
012040' 062702  000012                          ADD     #10.,R2                 ;ALLOW FOR I/O AREAS & "END" STMNT
012044' 066702  000344                          ADD     RDIOSZ,R2
012050' 066702  000350                          ADD     WRIOSZ,R2
012054' 016701  167370                          MOV     CVMEND,R1               ;GET END OF FREE MEM ADR
012060' 020201                                  CMP     R2,R1                   ;ENOUGH ROOM SO FAR?
012062' 103402                                  BLO     5$
012064' 000167  000406                  2$:     JMP     CLDER1                  ;Y,N-CLDER1
012070' 160201                          5$:     SUB     R2,R1                   ;GET SIZE OF AVAILABLE MEM
012072' 010167  000146                          MOV     R1,CDVRSZ               ;STORE IT IN LINK INFO
012076' 026767  030474  167514                  CMP     CNONID+4,CMCDVN         ;THIS THE "NONE" DEV?
012104' 001035                                  BNE     20$                     ;Y,N-20$
012106' 022701  000120                          CMP     #DRTLTH+2,R1            ;ENOUGH ROOM FOR "NONE" DEV ROUT?
012112' 101364                                  BHI     2$                      ;Y,N-CLDER1
012114' 016700  000122                          MOV     CDVRST,R0               ;GET DEV ROUT AREA START ADR
012120' 012720  000120                          MOV     #DRTLTH+2,(R0)+         ;STORE "NONE" DEV ROUT LENGTH
012124' 012701  000022                          MOV     #DTOEAD/2,R1            ;GET # OF WORDS TO CLEAR
012130' 005020                          10$:    CLR     (R0)+                   ;CLEAR WORD FOR "NONE" DEV ROUT
012132' 005301                                  DEC     R1                      ;DECR WORD COUNT
012134' 001375                                  BNE     10$                     ;CNT = 0?  (Y,N-10$)
012136' 062700  000050                          ADD     #DRTEND-DEVIOB,R0       ;POINT AT END OF THE TBL
012142' 012710  177777                          MOV     #177777,(R0)            ;STORE TERMINATOR WORD
012146' 012740  000002                          MOV     #2,-(R0)                ;STORE TABLE POINTERS
012152' 012740  000004                          MOV     #4,-(R0)
012156' 012740  000006                          MOV     #6,-(R0)
012162' 012740  000010                          MOV     #10,-(R0)
012166' 012740  000012                          MOV     #12,-(R0)
012172' 012740  000014                          MOV     #14,-(R0)
012176' 000477                                  BR      CLDISU                  ;GO SET UP PROG TBL & DEV ROUT INFO
012200' 112767  000124  165572          20$:    MOVB    #'T,CLOCZ               ;STORE "TXXA??" CODE AND
012206' 116767  000516  165565                  MOVB    DVRID,CLOCZ+1           ;DVR I.D.
012214' 116767  000511  165560                  MOVB    DVRID+1,CLOCZ+2         ;CODE
012222' 112767  000101  165553                  MOVB    #'A,CLOCZ+3             ;IN DEVICE
012230' 012767  037477  165546                  MOV     #"??,CLOCZ+4            ;ROUT NAME AREA
012236' 004577  165610                          JSR     R5,@LOADVR              ;GO LOAD THIS DEV ROUT
012242' 000000                          CDVRST: .WORD   XXXX                    ;  DEV ROUT MEM START ADR
012244' 000000                          CDVRSZ: .WORD   XXXX                    ;  DEV ROUT AREA MAX SIZE
012246' 000234                                  .WORD   CLDER2-.                ;  DEV ERROR RETURN ADR
012250' 000226                                  .WORD   CLDER1-.                ;  INSUFF MEMORY ERR RET ADR
012252' 000002                                  .WORD   CLDNRT-.                ;  NORMAL RET ADR
012254' 010302                          CLDNRT: MOV     R3,R2                   ;POINT AT DEV ROUTINE
012256' 062702  000244                          ADD     #PTLGTH+DEVFWD,R2       ;FLAGWORD
```

```
3747  012262' 005712                          TST     (R2)                    ;IS IT ZERO?
3748  012264' 001402                          BEQ     30$                     ;N,Y-30$
3749  012266' 061202                          ADD     (R2),R2                 ;IT'S AN ADR - MAKE IT ABS
3750  012270' 004512                          JSR     R5,(R2)                 ;GO TO DEV'S LOAD HOUSEKEEPING
3751  012272' 032767  000004  021714  30$:    BIT     #LSI11,CSYSFW           ;WE ON AN LSI-11?
3752  012300' 001413                          BEQ     40$                     ;Y,N-40$
3753  012302' 010302                          MOV     R3,R2                   ;GET ADR OF DEV'S BUS REQUEST
3754  012304' 062702  000272                  ADD     #PTLGTH+DEVRPS,R2       ;POINT AT READ OR ONLY PRIORITY
3755  012310' 005712                          TST     (R2)                    ;IS IT 0?
3756  012312' 001406                          BEQ     40$                     ;N,Y-40$
3757  012314' 012722  000200                  MOV     #200,(R2)+              ;UNCOND. SET IT TO BR 4
3758  012320' 005712                          TST     (R2)                    ;IS THERE A WRITE PRIORITY?
3759  012322' 001402                          BEQ     40$                     ;Y,N-40$
3760  012324' 012712  000200                  MOV     #200,(R2)               ;SET IT TO BR 4 ALSO
3761  012330' 016700  165526          40$:    MOV     REALNM,R0               ;GET ADR OF FILE'S NAME
3762  012334' 012067  175630                  MOV     (R0)+,FLLDNM            ;MOVE NAME TO THE MSG
3763  012340' 012067  175626                  MOV     (R0)+,FLLDNM+2
3764  012344' 011067  175624                  MOV     (R0),FLLDNM+4
3765  012350' 004567  173650                  JSR     R5,CTYPE                ;ISSUE MSG WITH FILENAME
3766  012354' 000600                          .WORD   600
3767  012356' 175576                          .WORD   FLLDMG-.
3768  012360' 010446                          MOV     R4,-(SP)                ;SAVE R4
3769  012362' 010304                          MOV     R3,R4                   ;SET UP ADR OF DEV ROUT
3770  012364' 062704  000266                  ADD     #PTLGTH+DEVDRA,R4       ;DEV INFO
3771  012370' 004567  176576                  JSR     R5,CRQADR               ;GO REQUEST DEV INFO CHANGES
3772  012374' 012604                          MOV     (SP)+,R4                ;RESTORE R4
3773  012376' 017701  177640          CLDISU: MOV     @CDVRST,R1              ;GET DEV ROUT LENGTH
3774  012402' 066701  177634                  ADD     CDVRST,R1               ;POINT AT END OF DEV ROUT
3775  012406' 010163  000016                  MOV     R1,PRDIOA(R3)           ;SET UP ADR OF RDIO AREA
3776  012412' 062701                          ADD     (PC)+,R1                ;ADJ ADR TO WRIO
3777  012414' 000400                  RDIOSZ: .WORD   256.
3778  012416' 010163  000020                  MOV     R1,PWRIOA(R3)           ;STORE WRIO AREA ADR
3779  012422' 062701                          ADD     (PC)+,R1                ;INCR PAST WRIO AREA
3780  012424' 000400                  WRIOSZ: .WORD   256.
3781  012426' 010163  000022                  MOV     R1,PSRCST(R3)           ;STORE SOURCE STMNTS START ADR
3782  012432' 010167  167010                  MOV     R1,CVMST                ;STORE NEW FREE MEM START ADR
3783  012436' 010700                          MOV     PC,R0                   ;GET TABLE ADR OF ADR'S TO
3784  012440' 062700  000062                  ADD     #CLDTBL-.,R0            ;BE PLACED IN DEV ROUT
3785  012444' 010302                          MOV     R3,R2                   ;SET UP THEIR ADR IN DEV ROUT
3786  012446' 062702  000310                  ADD     #PTLGTH+DEVIOB,R2
3787  012452' 012001                  CLDLOP: MOV     (R0)+,R1                ;GET REL OR VIRTUAL ADR
3788                  001                     .IF DF MM
3789                                          BMI     CLDBAS                  ;PAGE 6 ADR? (N,Y-CLDBAS)
3790                  000                     .ENDC
3791  012454' 060701                          ADD     PC,R1                   ;ITS REL, MAKE IT ABSOLUTE
3792  012456' 010122                  CLDBAS: MOV     R1,(R2)+                ;STORE IT IN DEV ROUT
3793  012460' 005710                          TST     (R0)                    ;END OF TBL?
3794  012462' 001373                          BNE     CLDLOP                  ;Y,N-CLDLOP
3795                  001                     .IF DF MM
3796                                          TST     PTLGTH+DEVIW7(R3)       ;THIS DEV ROUT NEED UNIBUS MAP?
3797                                          BEQ     CLDEX                   ;Y,N-CLDEX
3798                                          BIS     #USEUBM,(R3)            ;SET ITS USE UNIBUS MAP FLG
3799                                          BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
3800                                          BNE     CRQUBM                  ;N,Y-CRQUBM
3801                                          JSR     PC,CKUBAD               ;GO CK IF PROG WILL RUN
3802                                          BR      CLDEX                   ;GO TO EXIT
```

```
3803                            CRQUBM: MOV     PSRCST(R3),R0           ;GET SIZE OF DEV ROUT + RDIO + WRIO
3804                                    BIC     #P4CONS,R0              ;RESET PAGE BIT
3805                                    MOV     #1,R1                   ;INITIALIZE # OF BITS FOR REGS
3806                                    MOV     R1,R2                   ;SET UP BINARY # OF REGS
3807                            50$:    SUB     #8192.,R0               ;DECR SIZE BY 4K WORDS
3808                                    BLE     55$                     ;FND AREA END? (N,Y-55$)
3809                                    ASL     R1                      ;ADD ANOTHER BIT TO # OF REGS
3810                                    INC     R1
3811                                    INC     R2                      ;ADD 1 TO BINARY CNT
3812                                    BR      50$                     ;GO CK FOR ANOTHER REG
3813                            55$:    MOVB    R1,PUBMAP+1(R3)         ;STORE # OF REGS NEEDED IN PROG TBL
3814                                    MOV     #31.,CLDUMV             ;SET UP MAX VALUE FOR THIS
3815                                    SUB     R2,CLDUMV               ;# OF REGS
3816                                    BIS     #60,R2                  ;CONVERT BIN # TO ASCII
3817                                    MOVB    R2,UMNREG               ;STORE # IN MSG
3818                            CLDAUB: JSR     R5,CTYPE                ;ASK FOR 1ST UNIBUS MAP REG
3819                                    .WORD   420
3820                                    .WORD   UMRGRQ-.
3821                                    .WORD   28.
3822                                    JSR     R5,@CTRD                ;ISSUE READ FOR REPLY
3823                                    .WORD   CCMDRD-.
3824                                    .WORD   30.
3825                                    JSR     R5,CSCNSP               ;SCAN READ DATA FOR REPLY
3826                                    BIT     #CSCFND,R2              ;<CR> OR <LF> ONLY ENTRY?
3827                                    BEQ     CLDUMT                  ;N,Y-CLDUMT
3828                                    JSR     R5,ASDECO               ;CONVERT IT TO PACKED DECIMAL
3829                                    .WORD   CLDER4-.
3830                                    JSR     R5,DECBIN               ;CONVERT PKED DEC TO BINARY
3831                                    .WORD   CLDER4-.
3832                                    CMP     R0,#7                   ;IS # < 7?
3833                                    BLO     CLDER5                  ;N,Y-CLDER5
3834                                    CMP     R0,(PC)+                ;IS # > THAN MAX VALUE?
3835                            CLDUMV: .WORD   30.
3836                                    BHI     CLDER5                  ;N,Y-CLDER5
3837                                    MOV     R0,-(SP)                ;SAVE BINARY REG #
3838                                    MOVB    PUBMAP+1(R3),R2         ;GET # OF REG BITS
3839                                    CLR     R1                      ;CLEAR HI BITS
3840                                    SUB     #6,R0                   ;ADJ REG # FOR BIT POSITION
3841                            60$:    DEC     R0                      ;SHIFTED BITS ENOUGH?
3842                                    BEQ     65$                     ;N,Y-65$
3843                                    ASL     R2                      ;SHIFT BITS ONE POSITION
3844                                    ROL     R1
3845                                    BR      60$                     ;GO CK BIT SHIFT CNT
3846                            65$:    BIT     R1,MAPMAP               ;ANY OF HIGH REGS IN USE?
3847                                    BNE     CLDER6                  ;N,Y-CLDER6
3848                                    BIT     R2,MAPMAP+2             ;HOW ABOUT LOW REGS?
3849                                    BNE     CLDER6                  ;N,Y-CLDER6
3850                            CLDUME: BIS     R1,MAPMAP               ;SET BITS FOR THIS PROG'S REGS
3851                                    BIS     R2,MAPMAP+2             ;IN UB MAP REG USAGE BIT MAP
3852                                    MOVB    (SP)+,PUBMAP(R3)        ;STORE 1ST REG BIN # IN PROG TBL
3853                                    BR      CLDEX                   ;GO TO EXIT
3854                            CLDUMT: MOV     #7,-(SP)                ;INITIALIZE 1ST REG #
3855                                    MOVB    PUBMAP+1(R3),R2         ;GET # OF REG BITS
3856                                    CLR     R1
3857                                    MOV     #24.,R0                 ;SET UP MAX # OF SHIFTS ALLOWED
3858                                    CMP     R2,#3                   ;HOW MANY REGS NEEDED?
```

```
                                        BLO     75$                     ;BR IF 1
                                        BEQ     70$                     ;BR IF 2
                                        DEC     R0                      ;ADJ FOR 3 REGS
                                70$:    DEC     R0                      ;ADJ FOR 2 REGS
                                75$:    BIT     R1,MAPMAP               ;BITS IN USE IN HI REGS?
                                        BNE     80$                     ;N,Y-80$
                                        BIT     R2,MAPMAP+2             ;BITS IN USE IN LOW REGS?
                                        BEQ     90$                     ;Y,N-90$
                                80$:    DEC     R0                      ;DONE ALL SHIFTS POSSIBLE?
                                        BEQ     CLDER7                  ;N,Y-CLDER7
                                        INC     (SP)                    ;BUMP UP 1ST REG #
                                        ASL     R2                      ;SHIFT TO NEXT GROUP OF REGS
                                        ROL     R1
                                        BR      75$                     ;GO CK NEXT GROUP OF REGS
                                90$:    MOV     (SP),R0                 ;GET REG #
                                        JSR     R4,SREG                 ;SAVE R0 - R4
                                        JSR     PC,CPNAS1               ;CONVERT REG # TO DECIMAL ASCII
                                        MOV     R1,UMRGNM               ;STORE REG # IN MSG
                                        JSR     R5,CTYPE                ;DISPLAY # OF REG SELECTED
                                        .WORD   600
                                        .WORD   UMRGUD-.
                                        JSR     R4,RREG                 ;RESTORE R0 - R4
                                        BR      CLDUME                  ;GO UPDATE REG MAP
                        000             .ENDC
012464' 005063  000262          CLDEX:  CLR     PTLGTH+DEVIW7(R3)       ;RESET UB MAP USAGE FLAG
012470' 062705  000004          CLDER1: ADD     #4,R5                   ;INCR PAST ERR RET ADR'S
012474' 000205                  CLDER2: RTS     R5                      ;EXIT IN-LINE
012476' 061505                          ADD     (R5),R5                 ;SET UP ERR RET ADR
012500' 000205                          RTS     R5                      ;EXIT TO ERROR ADR
012502' 005725                  CLDER3: TST     (R5)+                   ;POINT AT DEV ERR RET ADR
012504' 000774                          BR      CLDER1                  ;GO TO ERROR EXIT
012506' 004567  173512          CLDER4: JSR     R5,CTYPE                ;ISSUE DATA CONV ERROR
012512' 000702                          .WORD   702
012514' 174073                          .WORD   CERM02-.
012516' 000167  177152                  JMP     CSZREQ                  ;GO RE-ISSUE MSG REQUEST
                        001             .IF DF MM
                                CLDER5: JSR     R5,CTYPE                ;ISSUE DATA CONV ERROR
                                        .WORD   702
                                        .WORD   CERM02-.
                                        BR      CLDERT                  ;GO RE-ISSUE UB MAP MSG
                                CLDER6: JSR     R5,CTYPE                ;ISSUE INV REG # MSG
                                        .WORD   712
                                        .WORD   CERM39-.
                                        BR      CLDERT                  ;GO RE-ISSUE UB MAP MSG
                                CLDER7: JSR     R5,CTYPE                ;ISSUE REGS IN USE ERROR MSG
                                        .WORD   603
                                        .WORD   CERM41-.
                                CLDERT: JMP     CLDAUB                  ;GO RE-ISSUE UB MAP MSG
                                        TST     (SP)+                   ;HSKP STACK
                                        JSR     R5,CTYPE                ;ISSUE INSUF REGS AVAIL MSG
                                        .WORD   203
                                        .WORD   CERM40-.
                        000             .ENDC
```

```
3915
3916                    001                     .IF NDF MM
3917  012522' 022540                    CLDTBL: .WORD   CIOBSY-CLDBAS
3918  012524' 022302                            .WORD   CUPGER-CLDBAS
3919  012526' 023162                            .WORD   ULIST-CLDBAS
3920  012530' 023122                            .WORD   CLIST-CLDBAS
3921  012532' 026444                            .WORD   BINASC-CLDBAS
3922  012534' 026514                            .WORD   BTASLZ-CLDBAS
3923  012536' 026346                            .WORD   DECASC-CLDBAS
3924  012540' 021536                            .WORD   CSYSFW-CLDBAS
3925  012542' 026214                            .WORD   SETVEC-CLDBAS
3926  012544' 026236                            .WORD   CLRVEC-CLDBAS
3927  012546' 026256                            .WORD   TSTVEC-CLDBAS
3928  012550' 026306                            .WORD   RTNINT-CLDBAS
3929  012552' 026310                            .WORD   GETBYT-CLDBAS
3930  012554' 026314                            .WORD   PUTBYT-CLDBAS
3931  012556' 000000                            .WORD   0
3932                    000                     .ENDC
3933                    001                     .IF DF MM
3934                                    CLDTBL: .WORD   CIOBSY-PG6BA+P6CONS     ;ALL LABELS CONTAINED IN THIS
3935                                            .WORD   CUPGER-PG6BA+P6CONS     ;TABLE MUST HAVE A MEMORY
3936                                            .WORD   ULIST-PG6BA+P6CONS      ;ADDRESS WHICH IS HIGHER THAN
3937                                            .WORD   CLIST-CLDBAS            ;"CLDBAS".
3938                                            .WORD   BINASC-PG6BA+P6CONS
3939                                            .WORD   BTASLZ-PG6BA+P6CONS
3940                                            .WORD   DECASC-PG6BA+P6CONS
3941                                            .WORD   CSYSFW-PG6BA+P6CONS
3942                                            .WORD   SETVEC-PG6BA+P6CONS
3943                                            .WORD   CLRVEC-PG6BA+P6CONS
3944                                            .WORD   TSTVEC-PG6BA+P6CONS
3945                                            .WORD   RTNINT-PG6BA+P6CONS
3946                                            .WORD   GETBYT-PG6BA+P6CONS
3947                                            .WORD   PUTBYT-PG6BA+P6CONS
3948                                            .WORD   0
3949                    000                     .ENDC
```

```
                                .SBTTL  VALIDATE MODEL NAME & DEV #'S S/R

                        ;*****************************************************************
                        ;
                        ;  THIS S/R SCANS THE KEYBOARD READ-IN AREA AND EXPECTS TO
                        ;  FIND A 4 CHARACTER ASCII MODEL NAME.  FOR THE 'ASSIGN' CMND
                        ;  ENTRY POINT (CVLMD1), THE MODEL NAME HAS ALREADY BEEN SCANNED
                        ;  BY THE CCFMT3 ROUTINE.  THE MODEL NAME WILL BE LOOKED UP IN THE
                        ;  VALID MODEL NAME TABLE AND ITS MODEL CODE WORD WILL BE EXTRACTED
                        ;  AND STORED.  NEXT, THE KEYBOARD DATA WILL BE SCANNED FOR UP TO
                        ;  SIXTEEN DEVICE #'S.  IF ANY ARE FOUND, THEY WILL BE STORED
                        ;  AT "CMCDVN" ALSO.
                        ;
                        ;  LINKAGE:     JSR     R5,CVLMD1       "ASSIGN" S/R CALL
                        ;               JSR     R5,CVLMD2       "ENTER" S/R CALL
                        ;
                        ;  DESTROYS REGISTERS:  R0,R1,R2
                        ;
                        ;*****************************************************************

012560' 052767  020000  166656  CVLMD1: BIS     #CVLFLG,CRFLGW          ;SET THE TRIALING OPERAND FLAG
012566' 000412                          BR      CVLCOM                  ;GO TO COMMON PROCESSING

012570' 004567  175422          CVLMD2: JSR     R5,CSCNSP               ;SCAN FOR THE MDL NAME
012574' 032702  000010                  BIT     #CSCFND,R2              ;WAS ANY DATA FOUND?
012600' 001002                          BNE     5$                      ;N,Y-5$
012602' 000167  000402                  JMP     CVLER1                  ;GO ISSUE ERROR MSG
012606' 042767  020000  166630  5$:     BIC     #CVLFLG,CRFLGW          ;RESET TRAILING OPERAND FLAG
012614' 022701  000004          CVLCOM: CMP     #4,R1                   ;MDL NAME = 4 CHAR'S?
012620' 001177                          BNE     CVLER2                  ;Y,N-CVLER2
012622' 010700                          MOV     PC,R0                   ;SET UP MDL NAME TBL START ADR
012624' 062700  027746                  ADD     #CMDLNM-.,R0
012630' 016701  166742          10$:    MOV     CDSTAD,R1               ;GET KEYBOARD DATA START ADR
012634' 005760  000004                  TST     4(R0)                   ;END OF THE MDL NAME TABLE?
012640' 100567                          BMI     CVLER2                  ;N,Y-CVLER2
012642' 012702  000004                  MOV     #4,R2                   ;SET UP COMPARE COUNT
012646' 122021                  20$:    CMPB    (R0)+,(R1)+             ;DATA CHAR = TBL CHAR?
012650' 001403                          BEQ     30$                     ;N,Y-30$
012652' 060200                          ADD     R2,R0                   ;POINT TO NXT TBL ENTRY
012654' 005200                          INC     R0
012656' 000764                          BR      10$                     ;GO CK NXT ENTRY
012660' 005302                  30$:    DEC     R2                      ;DECR CHAR CNT
012662' 001371                          BNE     20$                     ;CNT = 0? (Y,N-20$)
012664' 010346                          MOV     R3,-(SP)                ;SAVE WORK REG'S
012666' 010446                          MOV     R4,-(SP)
012670' 010703                          MOV     PC,R3                   ;SET UP STORAGE ADR
012672' 062703  166726                  ADD     #CMCDVN-.,R3
012676' 011023                          MOV     (R0),(R3)+              ;STORE MODEL CODE BYTES
012700' 005023                          CLR     (R3)+                   ;RESET DEV # PNTR & CURR DEV #
012702' 116002  000001                  MOVB    1(R0),R2                ;GET DVR I.D. CODE DISPL VALUE
012706' 032702  000001                  BIT     #1,R2                   ;THIS A DEFAULT MDL NAME?
012712' 001007                          BNE     DFLTMN                  ;N,Y-DFLTMN
012714' 010704                          MOV     PC,R4                   ;ADD IN TBL START ADR
012716' 062704  027652                  ADD     #DIDTAD-.,R4
```

```
4007  012722' 061402                        ADD     (R4),R2
4008  012724' 060402                        ADD     R4,R2
4009  012726' 011227                        MOV     (R2),(PC)+      ;SAVE DVR'S I.D.
4010  012730' 000000              DVRID:    .WORD   XXXX            ;DEV ROUT I.D. CODE WORD
4011  012732' 012704 000020       DFLTMN:   MOV     #16.,R4         ;SET UP DEV # CNT
4012  012736' 012727 177777                 MOV     #-1,(PC)+       ;RESET FIRST DEV # OF STRING
4013  012742' 000000              DVNUM1:   .WORD   0               ;FIRST DEV # OF A STRING
4014  012744' 005767 175244       35$:      TST     CSCFWD          ;ANOTHER OPERAND IN CMND?
4015  012750' 100466                        BMI     60$             ;Y,N-60$
4016  012752' 004567 175250                 JSR     R5,CSCNMS       ;SCAN FOR DEV #
4017  012756' 122777 000124 166612          CMPB    #'T,@CDSTAD     ;FIRST CHAR OF WORD = "T"?
4018  012764' 001467                        BEQ     80$             ;N,Y-80$
4019  012766' 004567 027356                 JSR     R5,ASDECO       ;CONVERT IT TO BINARY
4020  012772' 000236                        .WORD   CVLER3-.
4021  012774' 004567 027116                 JSR     R5,DECBIN
4022  013000' 000230                        .WORD   CVLER3-.
4023  013002' 016746 166612                 MOV     CMCDVN,-(SP)    ;GET DEVICE'S MODEL CODE
4024  013006' 042716 000366                 BIC     #366,(SP)       ;RESET I.D. BITS
4025  013012' 022627 023000                 CMP     (SP)+,#23000    ;THIS A TM02/TU16?
4026  013016' 001004                        BNE     38$             ;Y,N-38$
4027  013020' 020027 000115                 CMP     R0,#77.         ;DEV # > 77.?
4028  013024' 101101                        BHI     CVLER3          ;N,Y-CVLER3
4029  013026' 000403                        BR      40$             ;GO STORE THE #
4030  013030' 020027 000017       38$:      CMP     R0,#15.         ;DEV # > 15?
4031  013034' 101075                        BHI     CVLER3          ;N,Y-CVLER3
4032  013036' 005704              40$:      TST     R4              ;ALREADY DO 16 DEV #'S?
4033  013040' 001741                        BEQ     35$             ;N,Y-35$
4034  013042' 016702 177674                 MOV     DVNUM1,R2       ;GET 1ST DEV # OF A STRING
4035  013046' 100416                        BMI     50$             ;WAS THERE ONE? (Y,N-50$)
4036  013050' 120200              42$:      CMPB    R2,R0           ;1ST # = 2ND #?
4037  013052' 001410                        BEQ     48$             ;N,Y-48$
4038  013054' 103402                        BLO     44$             ;1ST # < 2ND #? (N,Y-44$)
4039  013056' 005302                        DEC     R2              ;SUB 1 FROM 1ST #
4040  013060' 000401                        BR      46$             ;GO STORE IT
4041  013062' 005202              44$:      INC     R2              ;ADD 1 TO 1ST #
4042  013064' 110223              46$:      MOVB    R2,(R3)+        ;STORE DEV #
4043  013066' 005304                        DEC     R4              ;DECR DEV # COUNT
4044  013070' 001725                        BEQ     35$             ;DONE 16 #'S? (N,Y-35$)
4045  013072' 000766                        BR      42$             ;GO CK IF FINISHED
4046  013074' 012767 177777 177640 48$:     MOV     #-1,DVNUM1      ;RESET FIRST DEV #
4047  013102' 000402                        BR      55$             ;GO CK THIS # FOR START OF A STRING
4048  013104' 005304              50$:      DEC     R4              ;DECR DEV # CNT
4049  013106' 110023                        MOVB    R0,(R3)+        ;MOVE CHAR TO STORAGE
4050  013110' 032767 000040 175076 55$:     BIT     #CSCTMS,CSCFWD  ;MINUS SIGN AFTER DEV #?
4051  013116' 001712                        BEQ     35$             ;Y,N-35$
4052  013120' 010067 177616                 MOV     R0,DVNUM1       ;SAVE 1ST DEV # OF STRING
4053  013124' 000707                        BR      35$             ;GO CK FOR ANOTHER DEV #
4054  013126' 004567 000022       60$:      JSR     R5,CVLFIL       ;FILL DEV #'S & CK TRAILING FLAG
4055  013132' 001044                        BNE     CVLER4          ;TRAILING FLAG SET? (N,Y-CVLER4)
4056  013134' 116767 166464 166461 70$:     MOVB    CMCDVN+4,CMCDVN+3 ;SET CURR DEV # TO FIRST DEV #
4057  013142' 000205                        RTS     R5              ;EXIT IN-LINE
4058  013144' 004567 000004       80$:      JSR     R5,CVLFIL       ;FILL DEV #'S & CK TRAILING FLG
4059  013150' 001371                        BNE     70$             ;TRAILING FLG SET? (N,Y-70$)
4060  013152' 000430                        BR      CVLR3A          ;GO REPORT INV UNIT #
4061
4062
```

```
4063  013154' 005704                  CVLFIL: TST     R4                      ;DEV # CNT = 0?
4064  013156' 001404                          BEQ     100$                    ;N,Y-100$
4065  013160' 112723  000377          90$:    MOVB    #377,(R3)+              ;MOV FILL CHAR TO DEV # STORAGE
4066  013164' 005304                          DEC     R4                      ;DECR DEV # CNT
4067  013166' 001374                          BNE     90$                     ;CNT = 0? (Y,N-90$)
4068  013170' 012600                  100$:   MOV     (SP)+,R0
4069  013172' 012604                          MOV     (SP)+,R4                ;RESTORE WORK REG'S
4070  013174' 012603                          MOV     (SP)+,R3
4071  013176' 010046                          MOV     R0,-(SP)
4072  013200' 032767  020000  166236          BIT     #CVLFLG,CRFLGW          ;TEST TRAILING OPERAND FLAG
4073  013206' 000205                          RTS     R5                      ;EXIT IN-LINE
4074
4075
4076  013210' 004567  173010          CVLER1: JSR     R5,CTYPE                ;ISSUE NO MDL NAME ERROR
4077  013214' 000043                          .WORD   043
4078  013216' 173617                          .WORD   CERM18-.
4079  013220' 004567  173000          CVLER2: JSR     R5,CTYPE                ;ISSUE INV MDL NAME ERROR
4080  013224' 000113                          .WORD   113
4081  013226' 173607                          .WORD   CERM18-.
4082  013230' 012604                  CVLER3: MOV     (SP)+,R4                ;RESTORE R3 & R4
4083  013232' 012603                          MOV     (SP)+,R3
4084  013234' 004567  172764          CVLR3A: JSR     R5,CTYPE                ;GO REPORT INV UNIT #
4085  013240' 000113                          .WORD   113
4086  013242' 173564                          .WORD   CERM17-.
4087  013244' 012605                  CVLER4: MOV     (SP)+,R5                ;CLEAR REG R5
4088  013246' 000167  171536                  JMP     CSANPN                  ;GO ISSUE NO PROG # MSG
```

```
                                .SBTTL  USER PROGRAM FLAGWORD MODIFICATION S/R'S

                        ;*****************************************************************
                        ;
                        ;  THESE SUBROUTINES PROVIDE QUICK AND EASY ACCESS TO THE
                        ;  FLAGWORD CONTAINED IN A USER PROGRAM AREA. THEY ARE
                        ;  REQUIRED FOR THE EXECUTIVE IN THE MEMORY MANAGEMENT
                        ;  VERSION. EQUIVALENT S/R'S ARE INCLUDED IN THE NON-MEMORY
                        ;  MANAGEMENT VERSION FOR COMPATIBILITY PURPOSES.
                        ;
                        ;*****************************************************************


                                ; SET BITS IN THE USER'S FLAGWORD
                                ;
                                ;       JSR     R5,UFWSET       S/R CALL
                                ;       .WORD   BITMSK          BITS TO BE SET
                                ;       .WORD   ADR-.           REL ADR OF WD THAT
                                ;                               CONTAINS USER FLGWD ADR.
                                ;                               IF MM, CONTAINS HIGH 16 BITS.
                                ; DESTROYS R1

            001                 .IF DF MM
                                .IFTF
013252' 012567  000006  UFWSET: MOV     (R5)+,10$               ;GET BIT MASK
013256' 010501                  MOV     R5,R1                   ;GET REL ADR AND MAKE
013260' 062501                  ADD     (R5)+,R1                ;IT ABS
                                .IFT
                                JSR     PC,UFWSUV               ;SET UP A VIRTUAL ADR
                                BIS     (PC)+,(R1)              ;SET SPECIFIED BITS
                                .IFF
013262' 052731                  BIS     (PC)+,@(R1)+            ;SET SPECIFIED BITS
                                .IFTF
013264' 000000          10$:    .WORD   XXXX
013266' 000417                  BR      UFWMEX                  ;GO TO EXIT
            000                 .ENDC


                                ; CLEAR BITS IN THE USER'S FLAGWORD
                                ;
                                ;       JSR     R5,UFWCLR       S/R CALL
                                ;       .WORD   BITMSK          BITS TO BE CLEARED
                                ;       .WORD   ADR-.           REL ADR OF WD THAT
                                ;                               CONTAINS USER FLGWD ADR.
                                ;                               IF MM, CONTAINS HIGH 16 BITS.
                                ; DESTROYS R1

            001                 .IF DF MM
                                .IFTF
013270' 012567  000006  UFWCLR: MOV     (R5)+,10$               ;GET BIT MASK
013274' 010501                  MOV     R5,R1                   ;GET REL ADR AND MAKE
013276' 062501                  ADD     (R5)+,R1                ;IT ABS
                                .IFT
                                JSR     PC,UFWSUV               ;SET UP A VIRTUAL ADR
                                BIC     (PC)+,(R1)              ;CLEAR SPECIFIED BITS
                                .IFF
```

```
 4146  013300' 042731                    BIC     (PC)+,@(R1)+            ;CLEAR SPECIFIED BITS
 4147                                    .IFTF
 4148  013302' 000000              10$:  .WORD   XXXX
 4149  013304' 000410                    BR      UFWMEX                  ;GO TO EXIT
 4150                                    .ENDC
```

```
                                        ; TEST BITS IN THE USER'S FLAGWORD
                                        ;
                                        ;       JSR     R5,UFWTST       S/R CALL
                                        ;       .WORD   BITMSK          BITS TO BE TESTED
                                        ;       .WORD   ADR-.           REL ADR OF WD THAT
                                        ;                               CONTAINS USER FLGWD ADR.
                                        ;                               IF MM, CONTAINS HIGH 16 BITS.
                                        ;       BR      LABEL           EXECUTED IF BITS NOT SET (BEQ)
                                        ;
                                        ; DESTROYS R1

            001                         .IF DF MM
                                        .IFTF
013306' 012567  000006          UFWTST: MOV     (R5)+,10$               ;GET BIT MASK
013312' 010501                          MOV     R5,R1                   ;GET REL ADR AND MAKE
013314' 062501                          ADD     (R5)+,R1                ;IT ABS
                                        .IFT
                                        JSR     PC,UFWSUV               ;SET UP A VIRTUAL ADR
                                        BIT     (PC)+,(R1)              ;TEST SPECIFIED BITS
                                        .IFF
013316' 032731                          BIT     (PC)+,@(R1)+            ;TEST SPECIFIED BITS
                                        .IFTF
013320' 000000                  10$:    .WORD   XXXX
013322' 001401                          BEQ     UFWMEX                  ;ANY BITS SET? (Y,N-UFWMEX)
013324' 005725                          TST     (R5)+                   ;BYPASS THE BR INST
                                        .IFF
013326' 000205                  UFWMEX: RTS     R5                      ;EXIT IN-LINE
                                        .IFT
                                UFWMEX: MOV     (SP)+,@#KPAR6           ;RESTORE KERNEL'S PAR 6
                                        RTS     R5                      ;EXIT IN-LINE

                                        ;SET UP VIRTUAL ADR OF UFW

                                UFWSUV: MOV     (SP),-(SP)              ;MAKE ROOM ON THE STK
                                        MOV     @#KPAR6,2(SP)           ;SAVE KERNEL'S PAR 6
                                        JSR     R4,SREG                 ;SAVE REG'S R0 - R4
                                        MOV     PC,R2                   ;GET ADR OF S/R LINK WD
                                        ADD     #20$-.,R2
                                        SUB     R2,R1                   ;MAKE ADR REL AGAIN
                                        MOV     R1,(R2)                 ;STORE IT
                                        JSR     R5,G22BAD               ;CONVERT IT TO A 22 BIT ADR
                                20$:    .WORD   XXXX
                                        JSR     PC,CVTVAD               ;CONVERT IT TO A VIRTUAL ADR
                                        MOV     R1,2(SP)                ;PUT ADR IN R1 ON THE STK
                                        JSR     R4,RREG                 ;RESTORE REGISTERS
                                        RTS     PC                      ;EXIT IN-LINE
            000                         .ENDC
            001                         .IF DF MM
                                        .PAGE
                                        .SBTTL  CHECK FOR UP TO A 3 PAGE MEMORY AREA

                                ;***************************************************************
                                ;
                                ;  USING THE CONTENTS OF R1 AS THE HIGH 16 BITS OF A 22 BIT ADDRESS,
```

```
                        ;  THIS SUBROUTINE WILL DETERMINE THE AMOUNT OF AVAILABLE MEMORY
                        ;  FROM THIS ADDRESS UP TO THE NEXT USER PROGRAM OR THE END OF
                        ;  MEMORY. THE SIZE OF THIS AREA (MAX OF 12K WORDS) IS RETURNED IN
                        ;  R2 AS A COUNT OF 32 WORD BLOCKS. AN ERROR WILL BE REPORTED IF
                        ;  THE MEMORY AREA IS ALREADY OCCUPIED AND R2 WILL BE SET TO 0.
                        ;
                        ;  LINKAGE:     JSR     PC,CKF3PG       S/R CALL
                        ;
                        ;  ENTRY:       R1 = HIGH 16 BIT ADR OF AREA TO BE CHECKED
                        ;
                        ;  EXIT:        R2 = AREA LENGTH IN 32 WORD BLOCKS
                        ;
                        ;  DESTROYS R2
                        ;
                        ;*************************************************************************

                        CKF3PG: JSR     R4,SREG                 ;SAVE REGS 0 - 4
                                MOV     PC,R0                   ;SET UP CNTRL ROUT TBL ADR
                                ADD     #CRTBL-2-.,R0
                                MOV     R1,R3                   ;MOVE AREA START ADR TO R3
                                MOV     R3,R4                   ;SET AREA END ADR TO AREA
                                ADD     #600,R4                 ;START + 12K WORDS
                                MOV     CRMEND,R2               ;GET REAL MEM END ADR
                                INC     R2                      ;ADJUST IT
                                CMP     R4,R2                   ;IS END ADR OUT OF MEM?
                                BLOS    10$                     ;Y,N-10$
                                MOV     R2,R4                   ;USE END OF MEM AS AREA END ADR
                        10$:    TST     (R0)+                   ;POINT TO NXT TBL ENTRY
                                TST     (R0)                    ;THIS PROG ENTRY EMPTY?
                                BEQ     10$                     ;N,Y-10$
                                CMP     (R0),#177777           ;END OF THE TBL?
                                BEQ     50$                     ;N,Y-50$
                                MOV     CRTEXT(R0),R2           ;GET PROG'S LGTH IN 32 WD BLKS
                                BIC     #FIXADR,R2              ;RESET FIX ADR FLG IF THERE
                                ADD     (R0),R2                 ;GET PROG'S END ADR
                                CMP     R3,(R0)                 ;AREA START ADR VS. PROG'S START ADR
                                BLO     20$                     ;BR IF AREA ADR IS LOWER
                                BHI     30$                     ;BR IF AREA ADR IS HIGHER
                                CMP     R0,CURCTE               ;THIS SAME PROG SLOT?
                                BEQ     10$                     ;N,Y-10$
                                BR      40$                     ;GO TO ERROR POINT
                        20$:    CMP     R4,(R0)                 ;AREA END ADR =< PROG'S START ADR?
                                BLOS    10$                     ;N,Y-10$
                                MOV     (R0),R4                 ;USE PROG'S ST ADR AS NEW AREA END ADR
                                BR      10$                     ;GO CK NXT ENTRY
                        30$:    CMP     R3,R2                   ;AREA START ADR => PROG'S END ADR?
                                BHIS    10$                     ;N,Y-10$
                        40$:    MOV     R3,R4                   ;SET AREA END ADR TO AREA START ADR
                                JSR     R5,CTYPE                ;ISSUE "AREA ALREADY OCCUPIED"
                                .WORD   402                     ;ERROR MSG
                                .WORD   CERM34-.
                        50$:    SUB     R3,R4                   ;GET LGTH OF THE AREA
                                MOV     R4,4(SP)                ;STORE IT IN R2 ON THE STK
                                JSR     R4,RREG                 ;RESTORE REGS 0 - 4
                                RTS     PC                      ;EXIT IN-LINE
                                .PAGE
```

```
                        .SBTTL  MOVE USER PROGRAMS UP AND DOWN S/R'S

                ;*****************************************************************
                ;
                ;  THE FOLLOWING TWO SUBROUTINES ARE USED TO MOVE USER PROGRAMS IN
                ;  MEMORY. THE ADDRESSES SUPPLIED ARE THE HIGH 16 BITS OF 22 BIT
                ;  ADDRESSES AND THE PROGRAM'S SIZE IS SPECIFIED IN THE NUMBER OF
                ;  32 WORD BLOCKS. THE MOVE UP S/R, WHICH IS USED TO RELOCATE
                ;  PROGRAMS UPWARDS IN MEMORY, STARTS AT THE END OF THE PROGRAM
                ;  AND MOVES CONSECUTIVELY LOWER BLOCKS. THE MOVE DOWN S/R, WHICH
                ;  IS USED TO RELOCATE PROGRAMS DOWNWARD IN MEMORY, STARTS AT THE
                ;  BEGINNING OF THE PROGRAM AND MOVES CONSECUTIVELY HIGHER BLOCKS.
                ;
                ;*****************************************************************

                        ;MOVE USER PROGRAM UP

                                ;JSR     PC,MOVPGU       S/R CALL
                                ;R2 = # OF 32 WORD BLOCKS
                                ;R3 = "FROM" 16 HI BIT ADR
                                ;R4 = "TO" 16 HI BIT ADR
                                ;DESTROYS R0,R1,R2,KPAR6 - ALTERS R3,R4

MOVPGU: MOV     R2,-(SP)                ;SAVE PROG LENGTH
        ADD     R2,R3                   ;POINT TO END OF "FROM" AREA
        DEC     R3
10$:    MOV     R3,MOVADR               ;STORE "FROM" HI 16 BIT ADR
        JSR     PC,GET32W               ;GET 32 WDS OF PROG
        MOV     R4,MOVADR               ;STORE "TO" HI 16 BIT ADR
        JSR     PC,PUT32W               ;PUT 32 WDS IN NEW AREA
        DEC     (SP)                    ;DECR PROG LENGTH CNT
        BEQ     MOVEX                   ;CNT = 0? (N,Y-MOVEX)
        DEC     R3                      ;DECR PROG AREA ADR'S
        DEC     R4
        BR      10$                     ;GO DO NXT 32 WDS
MOVEX:  TST     (SP)+                   ;CLEAN UP STACK
        RTS     PC                      ;EXIT IN-LINE


                        ;MOVE USER PROGRAM DOWN

                                ;JSR     PC,MOVPGD       S/R CALL
                                ;R3 = # OF 32 WD BLKS
                                ;R4 = ADR OF "FROM" HI 16 BIT ADR
                                ;R5 = "TO" HI 16 BIT ADR
                                ;DESTROYS R0,R1,R2,R3,KPAR6 - ALTERS (R4),R5

MOVPGD: MOV     R5,-(SP)                ;SAVE PROG'S NEW AREA ADR
30$:    MOV     (R4),MOVADR             ;STORE "FROM" HI 16 BIT ADR
        JSR     PC,GET32W               ;GET 32 WDS OF PROG
        MOV     R5,MOVADR               ;STORE "TO" HI 16 BIT ADR
        JSR     PC,PUT32W               ;PUT 32 WDS IN NEW AREA
        INC     (R4)                    ;ADD 1 TO "FROM" ADR
        INC     R5                      ;ADD 1 TO "TO" ADR
        DEC     R3                      ;DONE ALL 32 WD BLKS?
        BNE     30$                     ;Y,N-30$
```

```
4320                                    MOV     (SP)+,(R4)              ;STORE PROG'S NEW ADR
4321                                    RTS     PC                      ;EXIT IN-LINE
4322
4323
4324                            GET32W: MOV     #012120,GPINST          ;SET UP "MOV (R1)+,(R0)+" INST
4325                                    BR      GP32CM                  ;GO TO COM PROC
4326
4327                            PUT32W: MOV     #012021,GPINST          ;SET UP "MOV (R0)+,(R1)+" INST
4328                            GP32CM: JSR     R5,G22BAD               ;GET ADR AS 22 BITS
4329                                    .WORD   MOVADR-.
4330                                    JSR     PC,CVTVAD               ;SET UP A VIRTUAL ADR IN R1
4331                                    MOV     PC,R0                   ;SET UP WORK BUFFER ADR
4332                                    ADD     #USTKWK-.,R0
4333                                    MOV     #32.,R2                 ;SET UP WORD CNT
4334                            GPINST: NOP                             ;GET OR PUT A WORD
4335                                    DEC     R2                      ;DECR WD CNT
4336                                    BNE     GPINST                  ;CNT = 0? (Y,N-GPINST)
4337                                    RTS     PC                      ;EXIT IN-LINE
4338
4339                            MOVADR: .WORD   0                       ;*;HI 16 BIT ADR OF MEM BLK
4340                                    .PAGE
4341                                    .SBTTL  FIND LARGEST/LOWEST FREE MEMORY AREA
4342
4343                            ;*******************************************************************************
4344                            ;
4345                            ;  THIS SUBROUTINE SCANS THE UNUSED AREAS OF MEMORY LOOKING FIRST
4346                            ;  FOR A 12K WORD AREA. IF NONE ARE FOUND, THE NEXT LARGEST AREA
4347                            ;  IS SELECTED. IF TWO OR MORE AREAS ARE THE SAME SIZE AS THE LARGEST, THE
4348                            ;  AREA LOWEST IN MEMORY WILL BE SELECTED. THE STARTING ADDRESS OF THE AREA
4349                            ;  AND ITS SIZE WILL BE RETURNED IN REGISTERS. IF THE UNIBUS MAP IS NOT
4350                            ;  BEING USED, THE SELECTED AREA WILL BE WITHIN THE 18 BIT ADDRESSING AREA.
4351                            ;
4352                            ;  LINKAGE:     JSR     PC,FINDFM
4353                            ;
4354                            ;  EXIT:        R0 = HI 16 BITS OF AREA
4355                            ;               R1 = # OF 32 WD BLKS
4356                            ;
4357                            ;*******************************************************************************
4358
4359                            FINDFM: JSR     R4,SREG                 ;SAVE REGS 0 - 4
4360                                    MOV     PC,R0                   ;SET UP CNTRL ROUT TBL ADR
4361                                    ADD     #CRTBL-.,R0
4362                                    MOV     PC,R1                   ;SET UP WORK BUFFER ADR
4363                                    ADD     #USTKWK-.,R1
4364                                    MOV     R1,R2                   ;SAVE IT IN R2 FOR LATER
4365                            10$:    MOV     CRTEXT(R0),CRTEXT(R1)   ;MOVE TBL EXT WD TO WK BUF
4366                                    BIC     #FIXADR,CRTEXT(R1)      ;RESET FIXED ADR FLG IF THERE
4367                                    MOV     (R0)+,(R1)+             ;STORE USER PROG ADR
4368                                    CMP     #177777,-2(R0)          ;END OF CNTRL ROUT TBL?
4369                                    BNE     10$                     ;Y,N-10$
4370                                    MOV     R2,R1                   ;SET UP ADR OF 2ND
4371                                    ADD     #68.,R1                 ;WORK AREA
4372                                    MOV     R1,-(SP)                ;SAVE 2ND AREA ADR ON STK
4373                            20$:    MOV     CRMEND,-(SP)            ;PUT END OF MEM ADR ON STK
4374                                    CLR     R3                      ;RESET ENTRY FOUND FLG
4375                                    MOV     R2,R0                   ;POINT AT 1ST AREA IN WK BUF
```

```
4376                            30$:    TST     (R0)                    ;ENTRY EMPTY?
4377                                    BEQ     40$                     ;N,Y-40$
4378                                    BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
4379                                    BNE     35$                     ;N,Y-35$
4380                                    BIT     #BITS22,CSYSFW          ;USING 22 BIT ADRS?
4381                                    BEQ     35$                     ;Y,N-35$
4382                                    CMP     (R0),#10000             ;THIS PROG ABOVE 18 BIT ADRS?
4383                                    BLO     35$                     ;Y,N-35$
4384                                    CLR     (R0)                    ;RESET ITS ENTRY
4385                                    BR      40$                     ;GO CK NEXT SLOT
4386                            35$:    INC     R3                      ;SET THE ENTRY FND FLG
4387                                    CMP     (R0),(SP)               ;THIS LOWEST ADR SO FAR?
4388                                    BHIS    40$                     ;Y,N-40$
4389                                    MOV     (R0),(SP)               ;SAVE ITS ADR AS NEW LOWEST
4390                                    MOV     R0,R4                   ;SAVE ITS ENTRY ADR
4391                            40$:    TST     (R0)+                   ;POINT TO NXT TBL ENTRY
4392                                    CMP     #177777,(R0)            ;END OF THE TBL?
4393                                    BNE     30$                     ;Y,N-30$
4394                                    TST     (SP)+                   ;TAKE OLD ADR OFF STK
4395                                    TST     R3                      ;ANY PROG'S LEFT?
4396                                    BEQ     50$                     ;Y,N-50$
4397                                    MOV     CRTEXT(R4),CRTEXT(R1)   ;STORE THIS PROG'S INFO AS THE
4398                                    MOV     (R4),(R1)+              ;NEXT HIGHEST IN MEM
4399                                    CLR     (R4)                    ;RESET THIS PROG'S SLOT
4400                                    BR      20$                     ;GO LOOK FOR NXT HIGHEST PROG
4401                            50$:    MOV     #177777,(R1)+           ;STORE A TBL TERMINATOR
4402                                    MOV     (SP)+,R0                ;GET 2ND AREA ADR AGAIN
4403                                    MOV     CPAREA,R1               ;INIT PREV PROG END ADR
4404                                    MOV     R1,-(SP)                ;INIT PREV PROG-1 END ADR
4405                                    MOV     R1,-(SP)                ;INIT PREV PROG START ADR
4406                                    CLR     R4                      ;RESET LONGEST AREA CNT
4407                            60$:    CMP     #177777,(R0)            ;END OF SORTED PROG ADR TBL?
4408                                    BNE     70$                     ;Y,N-70$
4409                                    MOV     CRMEND,R2               ;GET END OF MEM ADR
4410                                    INC     R2                      ;ADJ IT
4411                                    BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
4412                                    BNE     65$                     ;N,Y-65$
4413                                    BIT     #BITS22,CSYSFW          ;USING 22 BIT ADRS?
4414                                    BEQ     65$                     ;Y,N-65$
4415                                    CMP     R2,#10000               ;MEM END > 18 BITS?
4416                                    BLOS    65$                     ;Y,N-65$
4417                                    MOV     #10000,R2               ;SET UP NEW MEM ADR
4418                                    CMP     R1,R2                   ;PREV PROG END AT 18 BITS OR ABOVE?
4419                                    BLO     65$                     ;Y,N-65$
4420                                    MOV     (SP),R2                 ;USE IT START ADR AS END OF MEM ADR
4421                                    MOV     2(SP),R1                ;GET END ADR OF PROG BEFORE IT
4422                            65$:    SUB     R1,R2                   ;SUB PREV PROG END ADR FROM IT
4423                                    CMP     R2,#600                 ;AREA => 3 PAGES IN SIZE?
4424                                    BHIS    100$                    ;N,Y-100$
4425                                    CMP     R2,R4                   ;IS IT LARGER THAN PREV LARGEST?
4426                                    BLOS    120$                    ;Y,N-120$
4427                                    MOV     R2,R4                   ;SET UP ITS LENGTH INSTEAD
4428                                    BR      110$                    ;GO TO FINAL PROCESSING
4429                            70$:    CMP     (R0),R1                 ;THIS PROG BUTT AGAINST PREV PROG?
4430                                    BNE     90$                     ;Y,N-90$
4431                            80$:    MOV     R1,2(SP)                ;SAVE AS PREV PROG-1 END ADR
```

```
4432                                    MOV     (R0),(SP)               ;SAVE AS PREV PROG START ADR
4433                                    MOV     (R0),R1                 ;USE ITS ADR + LGTH AS NEW
4434                                    ADD     CRTEXT(R0),R1           ;PREV PROG END ADR
4435                                    TST     (R0)+                   ;POINT TO NXT PROG SLOT
4436                                    BR      60$                     ;GO CK NXT PROG
4437                            90$:    MOV     (R0),R2                 ;GET ITS ADR
4438                                    SUB     R1,R2                   ;SUB PREV PROG END ADR FROM IT
4439                                    CMP     R2,#600                 ;OPEN AREA => 3 PAGES?
4440                                    BHIS    100$                    ;N,Y-100$
4441                                    CMP     R2,R4                   ;IS IT THE LARGEST SO FAR?
4442                                    BLOS    80$                     ;Y,N-80$
4443                                    MOV     R1,R3                   ;SAVE PREV PROG END ADR AS OPEN AREA ADR
4444                                    MOV     R2,R4                   ;SAVE ITS SIZE
4445                                    BR      80$                     ;GO CK FOR LARGER AREA
4446                            100$:   MOV     #600,R4                 ;SET AREA SIZE TO 3 PAGES
4447                            110$:   MOV     R1,R3                   ;USE PREV PROG END ADR AS START ADR
4448                            120$:   ADD     #4,SP                   ;REMOVE ADRS FROM STACK
4449                                    MOV     R3,(SP)                 ;STORE ADR IN R0 ON THE STK
4450                                    MOV     R4,2(SP)                ;STORE LGTH IN R1 ON THE STK
4451                                    JSR     R4,RREG                 ;RESTORE REGS 0 - 4
4452                                    RTS     PC                      ;EXIT IN-LINE
4453                                    .PAGE
4454                                    .SBTTL  SET UP MIDL'S AND FREE MEM'S ABSOLUTE ADRESSES
4455
4456                            ;*******************************************************************
4457                            ;
4458                            ;  THIS S/R WILL FIRST STORE THE 18/22 BIT ABSOLUTE ADDRESS OF
4459                            ;  "FREE" AND THEN CALCULATE AND STORE THE 18/22 BIT ABSOLUTE
4460                            ;  ADDRESS OF "MIDL". NEXT, THE PAR/PDR STORAGE AREAS FOR THE
4461                            ;  "FREE" AREA ARE TAILORED FOR FREE'S LOCATION AND LENGTH. IF
4462                            ;  THE UNIBUS MAP IS BEING USED, 18 BIT VIRTUAL ADDRESSES, WHICH
4463                            ;  ARE BASED UPON MAP REGISTERS 3 THRU 5, WILL ALSO BE STORED.
4464                            ;  THESE UNIBUS MAP REGISTERS WILL BE INITIALIZED WITH THE
4465                            ;  CORRECT ABSOLUTE ADDRESSES AT THIS TIME.
4466                            ;
4467                            ;  LINKAGE:     JSR     PC,SUMIFM
4468                            ;
4469                            ;  DESTROYS R0,R1,R2
4470                            ;
4471                            ;*******************************************************************
4472
4473                            SUMIFM: MOV     R3,-(SP)                ;SAVE R3 & R4
4474                                    MOV     R4,-(SP)
4475                                    JSR     R5,G22BAD               ;CONVERT FREE MEM START TO
4476                                    .WORD   CFMST-.                 ;22 BITS
4477                                    MOV     PC,R3                   ;GET ADR OF STORAGE AREA
4478                                    ADD     #FREEAD+2-.,R3
4479                                    MOV     R2,(R3)+                ;STORE FREE'S 18/22 BIT
4480                                    MOV     R0,(R3)+                ;ABS ADR
4481                                    BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
4482                                    BEQ     20$                     ;Y,N-20$
4483                                    CLR     (R3)+                   ;STORE FREE'S 18 BIT ADR BASED
4484                                    MOV     #P3CONS,(R3)+           ;UPON MAP REGS 3 THRU 5
4485                                    MOV     #3,R1                   ;SET UP LOOP CNT
4486                                    MOV     #UBMAP3,R4              ;INITIALIZE MAP REG ADR
4487                            10$:    MOV     R0,(R4)+                ;STORE ADR IN MAP REG
```

```
4488                                    MOV     R2,(R4)+
4489                                    ADD     #20000,R0               ;POINT TO NEXT 4K WORDS
4490                                    ADC     R2
4491                                    DEC     R1                      ;DONE 3 REGS?
4492                                    BNE     10$                     ;Y,N-10$
4493                                    SUB     #60000,R0               ;RE-ADJUST FREE'S ABS ADR
4494                                    SBC     R2
4495                            20$:    CLR     MIDADJ                  ;RESET ADJ FACTOR
4496                                    MOV     CFMEND,R1               ;GET END OF FREE MEM
4497                                    INC     R1                      ;ALLOW FOR LAST 32 WD BLK
4498                                    SUB     CFMST,R1                ;FIND FREE MEM SIZE
4499                                    MOV     R1,-(SP)                ;SAVE IT FOR LATER
4500                                    CLC                             ;DIVIDE IT BY 2
4501                                    ROR     R1
4502                                    BCC     30$                     ;BR IF EVEN # OF 32 WD BLKS
4503                                    MOV     #40,MIDADJ              ;STORE ADJ FACTOR
4504                            30$:    MOV     #6,R4                   ;CONVERT 32 WD CNT TO A
4505                            40$:    ASL     R1                      ;BYTE CNT
4506                                    DEC     R4
4507                                    BNE     40$
4508                                    BIS     (PC)+,R1                ;SET IN ADJ FACTOR
4509                            MIDADJ: .WORD   XXXX
4510                                    MOV     PC,R3                   ;SET UP ADR OF MIDL'S
4511                                    ADD     #MIDLAD-.,R3            ;STORAGE AREA
4512                                    MOV     #P3CONS,(R3)            ;STORE MIDL'S 16 BIT VIRTUAL ADR
4513                                    ADD     R1,(R3)+
4514                                    ADD     R1,R0                   ;ADD DISPL TO FREE'S ADR
4515                                    ADC     R2
4516                                    MOV     R2,(R3)+                ;STORE MIDL'S 18/22 BIT
4517                                    MOV     R0,(R3)+                ;ABS ADR
4518                                    BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
4519                                    BEQ     60$                     ;Y,N-60$
4520                                    CLR     (R3)+                   ;SET MIDL'S 18 BIT VIRT ADR BASED
4521                                    MOV     -8.(R3),(R3)            ;UPON MAP REGS 3 THRU 5
4522                            60$:    MOV     (SP)+,R1                ;GET FREE'S SIZE SAVED EARLIER
4523                                    MOV     CFMST,R3                ;GET FREE'S START HI 16 BITS
4524                                    MOV     PC,R4                   ;SET UP STORAGE AREA ADR
4525                                    ADD     #FMPAR3-.,R4
4526                            MAPEXT: MOV     #3,-(SP)                ;SET UP PAGE CNT
4527                            70$:    CMP     R1,#200                 ;SIZE < 4K WORDS?
4528                                    BLO     80$                     ;N,Y-80$
4529                                    MOV     #PDRCON,16.(R4)         ;STORE 4K PDR VALUE
4530                                    MOV     R3,(R4)+                ;STORE HI 16 BITS OF ADR
4531                                    ADD     #200,R3                 ;BUMP UP ADR BY 4K
4532                                    SUB     #200,R1                 ;DECREASE SIZE BY 4K
4533                                    DEC     (SP)                    ;DONE 3 PAGES?
4534                                    BNE     70$                     ;Y,N-70$
4535                                    BR      100$                    ;GO TO EXIT
4536                            80$:    TST     R1                      ;PARTIAL PAGE?
4537                                    BEQ     90$                     ;Y,N-90$
4538                                    DEC     R1                      ;ADJ # OF BLKS FOR PLF
4539                                    SWAB    R1                      ;SET UP PLF FIELD
4540                                    BIS     #6,R1                   ;SET IN ACF BITS
4541                                    MOV     R1,16.(R4)              ;STORE PARTIAL PAGE PDR
4542                                    MOV     R3,(R4)+                ;STORE ITS START ADR
4543                                    DEC     (SP)                    ;DONE 3 PAGES?
```

```
4544                            BEQ     100$                    ;N,Y-100$
4545                    90$:    CLR     16.(R4)                 ;RESET UNUSED PDR
4546                            CLR     (R4)+                   ;RESET UNUSED PAR
4547                            DEC     (SP)                    ;DONE 3 PAGES?
4548                            BNE     90$                     ;Y,N-90$
4549                    100$:   TST     (SP)+                   ;TAKE CNT OFF STK
4550                            MOV     (SP)+,R4                ;RESTORE R3 & R4
4551                            MOV     (SP)+,R3
4552                            RTS     PC                      ;EXIT IN-LINE
4553                            .PAGE
4554                            .SBTTL  MAP USER PROGRAM'S PAR & PDR REGISTER VALUES
4555
4556                    ;*******************************************************************
4557                    ;
4558                    ;  THIS SUBROUTINE WILL INITIALIZE THE PROGRAM TABLE STORAGE
4559                    ;  AREA FOR THIS PROGRAM'S PAR AND PDR REGISTERS. THE FOLLOWING
4560                    ;  ARE THE REGISTER ASSIGNMENTS:
4561                    ;
4562                    ;       PAR/PDR 0,1,2 = PROGRAM'S AREA
4563                    ;               3,4,5 = FREE MEMORY AREA
4564                    ;               6     = MPG'S SHARED CODE
4565                    ;               7     = I/O PAGE
4566                    ;
4567                    ;  USES PART OF THE "SUMIFM" S/R TO MAP PAR/PDR'S 0,1,2.
4568                    ;
4569                    ;  LINKAGE:     JSR     PC,MAPUPG
4570                    ;
4571                    ;  ENTRY:       R3 CONTAINS PROG TBL ADR
4572                    ;               R4 CONTAINS PROGRAM'S CONTROL ROUTINE TABLE ENTRY ADR
4573                    ;
4574                    ;  DESTROYS R0,R1,R2
4575                    ;
4576                    ;*******************************************************************
4577
4578                    MAPUPG: MOV     R3,-(SP)                ;SAVE R3 & R4
4579                            MOV     R4,-(SP)
4580                            MOV     PC,R0                   ;SET UP ADR OF VALUES FOR
4581                            ADD     #FMPAR3-.,R0            ;PAR'S & PDR'S 3 THRU 7
4582                            MOV     R3,R1                   ;POINT TO THEIR STORAGE AREA IN
4583                            ADD     #PUPARS+6,R1            ;THIS PROG'S TBL
4584                            MOV     #5,R2                   ;SET UP REG CNT
4585                    10$:    MOV     16.(R0),16.(R1)         ;STORE PDR VALUE
4586                            MOV     (R0)+,(R1)+             ;STORE PAR VALUE
4587                            DEC     R2                      ;DONE ALL REGS?
4588                            BNE     10$                     ;Y,N-10$
4589                            MOV     CRTEXT(R4),R1           ;GET PROG'S LGTH IN 32 WD BLKS
4590                            BIC     #FIXADR,R1              ;RESET FIXED ADR FLG IF THERE
4591                            MOV     R3,R4                   ;SET UP ADR OF STORAGE AREA IN
4592                            ADD     #PUPARS,R4              ;THIS PROG'S TBL
4593                            MOV     PFWADR(R3),R3           ;GET STARTING HI 16 BIT ADR
4594                            BR      MAPEXT                  ;USING PART OF SUMIFM S/R, DO MAPPING
4595             000            .ENDC
```

```
                                        .SBTTL  CONTROL ROUTINE SMALL S/R AREA

                                        ;***************
                                        ;
                                        ;  THIS S/R INITIALIZES THE CURRENT COMMAND DATA POINTER
                                        ;  TO THE START OF THE CONSOLE READ-IN AREA.
                                        ;
                                        ;  LINKAGE:     JSR     PC,CSUPTR
                                        ;
                                        ;***************

013330' 010746              CSUPTR: MOV     PC,-(SP)                ;RESET DATA PNTR TO FIRST
013332' 062716  166132              ADD     #CCMDRD-.,(SP)          ;CHAR OF READ IN AREA
013336' 012667  166120              MOV     (SP)+,CPNTR
013342' 000207                      RTS     PC                      ;EXIT IN-LINE


                                        ;***************
                                        ;
                                        ;  THIS S/R TESTS THE OCTAL ADDRESS IN R0 AND IF ODD,
                                        ;  SETS IT TO EVEN AND ISSUES A MSG INDICATING THIS FACT.
            001                         .IF DF MM
                                        ;  R2 CONTAINS HIGH 6 BITS OF THE OCTAL ADDRESS
            000                         .ENDC
                                        ;
                                        ;  LINKAGE:     JSR     R5,CKODAD
                                        ;
                                        ;  MAY DESTROY REGISTERS: R0,R1,R2
                                        ;
                                        ;***************

            001                         .IF DF MM
                                        .IFT
                                CKODAD: MOV     R2,-(SP)                ;SAVE ORG 22 BIT ADR
                                        MOV     R0,-(SP)
                                        .IFF
013344' 010046              CKODAD: MOV     R0,-(SP)                ;SAVE ORG ADR
            000                         .ENDC
013346' 032700  000001              BIT     #1,R0                   ;IS OCTAL ADR ODD?
013352' 001410                      BEQ     10$                     ;(Y,N-10$)
013354' 005316                      DEC     (SP)                    ;SET IT TO NXT LOWEST EVEN ADR
013356' 004567  025540              JSR     R5,BINAS8               ;CNVRT ORG ADR TO ASCII &
013362' 173753                      .WORD   CAAMSG-.                ;PUT IN MSG
013364' 004567  172634              JSR     R5,CTYPE                ;ISSUE ADR ADJUSTED MSG
013370' 000400                      .WORD   400
013372' 173743                      .WORD   CAAMSG-.
013374' 012600              10$:    MOV     (SP)+,R0                ;RESTORE FINAL ADR
            001                     .IF DF MM
                                    MOV     (SP)+,R2
            000                     .ENDC
013376' 000205                      RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;***************
                                        ;
                                        ;  THE ADDRESS CONTAINED AT THE STANDARD ERROR RETURN
                                        ;  ADDRESS STORAGE LOCATION "CERRET" WILL BE SET TO
                                        ;  "CENTPB" BY THIS S/R.
                                        ;
                                        ;  LINKAGE:     JSR     PC,CSUERR
                                        ;
                                        ;***************

013400' 010746                  CSUERR: MOV     PC,-(SP)                ;GET ADR OF ERROR RETURN POINT
013402' 062716  166754                  ADD     #CENTPB-.,(SP)
013406' 012667  166170                  MOV     (SP)+,CERRET            ;STORE IT IN ERROR REPORTING ROUT
013412' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;***************
                                        ;
                                        ;  THIS S/R MOVES THE 10 WORDS OF DEVICE INFO AT "CMCDVN"
                                        ;  TO THE PROGRAM TABLE POINTED TO BY R3.
                                        ;
                                        ;  LINKAGE:     JSR     R5,CSTDVI
                                        ;
                                        ;  DESTROYS REGISTERS: R0,R1,R2
                                        ;
                                        ;***************

013414' 010302                  CSTDVI: MOV     R3,R2                   ;SET UP PROG AREA WORK ADR
013416' 062702  000032                  ADD     #PMDLCD,R2              ;SET UP DEVICE INFO STORAGE ADR
013422' 010701                          MOV     PC,R1                   ;GET ADR OF DEVICE INFO
013424' 062701  166174                  ADD     #CMCDVN-.,R1
013430' 012700  000012                  MOV     #10.,R0                 ;SET UP WD CNT
013434' 012122                  20$:    MOV     (R1)+,(R2)+             ;MOVE DEV INFO TO PROG TBL
013436' 005300                          DEC     R0
013440' 001375                          BNE     20$
013442' 000205                          RTS     R5                      ;EXIT IN-LINE


                                        ;***************
                                        ;
                        001             .IF DF MM
                                        .IFF
                                        ;  THIS S/R VALIDATES THAT THE ADDRESS IN R4 IS EITHER
                                        .IFT
                                        ;  THIS S/R VALIDATES THAT THE ADDRESS IN R3 & R4 IS EITHER
                        000             .ENDC
                                        ;  WITHIN REAL MEMORY OR IN THE DEVICE REGISTERS' AREA.
                                        ;  IF NOT, AN ERROR IS REPORTED.
                                        ;
                                        ;  LINKAGE:     JSR     R5,CMEMCK
                                        ;
                                        ;***************

                        001             .IF DF MM
                                        .IFF
013444' 020467  166002          CMEMCK: CMP     R4,CRMEND               ;ADR HIGHER THAN REAL MEM ADR?
```

```
4706  013450' 101403                        BLOS    24$                     ;Y,N-24$
4707  013452' 020427  160000                CMP     R4,#160000              ;IN DEV REG RANGE?
4708  013456' 103401                        BLO     26$                     ;Y,N-26$
4709  013460' 000205                24$:    RTS     R5                      ;EXIT IN-LINE
4710  013462' 004567  172536        26$:    JSR     R5,CTYPE                ;GO REPORT OUT OF MEM ADR
4711  013466' 000003                        .WORD   003
4712  013470' 173225                        .WORD   CERM09-.
4713                                        .IFT
4714                                CMEMCK: JSR     R5,G22BAD               ;CONVERT REAL MEM END ADR TO
4715                                        .WORD   CRMEND-.                ;22 BITS
4716                                        BIS     #77,R0                  ;POINT TO END OF 32 WD AREA
4717                                        CMP     R3,R2                   ;COMPARE HIGH 6 BITS
4718                                        BHI     20$                     ; > MEM? (N,Y-20$)
4719                                        BLO     50$                     ; < MEM? (N,Y-50$)
4720                                        CMP     R4,R0                   ;COMPARE LOW 16 BITS
4721                                        BLOS    50$                     ; =< MEM? (N,Y-50$)
4722                                20$:    MOV     #3,R2                   ;SET UP HIGH BITS FOR 18 BITS
4723                                        BIT     #BITS22,CSYSFW          ;USING 22 BIT ADDRESSING?
4724                                        BEQ     30$                     ;Y,N-30$
4725                                        MOV     #77,R2                  ;SET UP HIGH BITS FOR 22 BITS
4726                                30$:    CMP     R3,R2                   ;COMPARE HIGH 6 BITS
4727                                        BNE     60$                     ; = I/O PAGE ADR? (Y,N-60$)
4728                                        CMP     R4,#160000              ;COMPARE LOW 16 BITS
4729                                        BLO     60$                     ; < I/O PAGE ADR? (N,Y-60$)
4730                                50$:    RTS     R5                      ;EXIT IN-LINE
4731
4732                                60$:    JSR     R5,CTYPE                ;GO REPORT OUT OF MEM
4733                                        .WORD   003
4734                                        .WORD   CERM09-.
4735            000                         .ENDC
4736
4737
4738                                ;***************
4739                                ;
4740                                ;  THIS S/R SCANS THE LAST CMND WORD BACKWARDS LOOKING
4741                                ;  FOR NON-DECIMAL CHARACTERS.
4742                                ;
4743                                ;  LINKAGE:     JSR     R5,CISDNM       S/R CALL
4744                                ;
4745                                ;  DESTROYS REGISTERS: R0,R1,R2
4746                                ;
4747                                ;  UPON EXIT:   R0 = RIGHT END ADR OF WORD
4748                                ;               R1 = # OF DECIMAL DIGITS
4749                                ;               R2 = ADR OF LEFTMOST DECIMAL DIGIT
4750                                ;
4751                                ;***************
4752
4753  013472' 010346                CISDNM: MOV     R3,-(SP)                ;SAVE WORK REG
4754  013474' 016700  166076                MOV     CDSTAD,R0               ;GET DATA STARTING ADR
4755  013500' 016703  166074                MOV     CDCNT,R3                ;GET DATA LENGTH
4756  013504' 060300                        ADD     R3,R0                   ;SET UP RIGHT HAND END ADR
4757  013506' 005300                        DEC     R0
4758  013510' 010002                        MOV     R0,R2                   ;MOVE IT TO WORK REG
4759  013512' 005001                        CLR     R1                      ;CLEAR CHAR CNT REG
4760  013514' 121227  000060        10$:    CMPB    (R2),#'0                ;CHAR IN OPER FROM 0 - 9?
4761  013520' 103407                        BLO     20$                     ;Y,N-20$
```

```
 4762  013522' 121227  000071          CMPB    (R2),#'9
 4763  013526' 101004                  BHI     20$
 4764  013530' 005302                  DEC     R2                      ;DECR DATA ADR
 4765  013532' 005201                  INC     R1                      ;INCR CHAR CNT
 4766  013534' 020103                  CMP     R1,R3                   ;SCANNED ENTIRE WORD?
 4767  013536' 001366                  BNE     10$                     ;Y,N-10$
 4768  013540' 005202          20$:    INC     R2                      ;POINT ADR AT PREV CHAR
 4769  013542' 012603                  MOV     (SP)+,R3                ;RESTORE WORK REG
 4770  013544' 000205                  RTS     R5                      ;EXIT IN-LINE
```

```
4772                                    ;***************
4773                                    ;
4774                                    ;  USING THE DEVICE MODEL CODE IN R1, FIND THE DEVICE'S NAME IN
4775                                    ;  THE VALID DEVICE NAME TABLE AND MOVE IT TO THE ADR SPECIFIED.
4776                                    ;
4777                                    ;  LINKAGE:     JSR     R5,FDVNAM       S/R CALL
4778                                    ;               .WORD   ADR-.           REL ADR TO PUT 4 CHAR NAME
4779                                    ;              R1 CONTAINS MODEL CODE WORD
4780                                    ;
4781                                    ;  DESTROYS REGISTERS: R0,R2
4782                                    ;
4783                                    ;***************
4784
4785  013546' 010700          FDVNAM: MOV     PC,R0                   ;SET UP ADR OF DEV NAME TBL
4786  013550' 062700  027022          ADD     #CMDLNM-.,R0
4787  013554' 010502                  MOV     R5,R2                   ;GET REL OUTPUT ADR &
4788  013556' 062502                  ADD     (R5)+,R2                ;MAKE IT ABS
4789  013560' 005760  000004  10$:    TST     4(R0)                   ;END OF THE TABLE?
4790  013564' 100406                  BMI     20$                     ;N,Y-20$
4791  013566' 020160  000004          CMP     R1,4(R0)                ;THIS THE MODEL CODE?
4792  013572' 001403                  BEQ     20$                     ;N,Y-20$
4793  013574' 062700  000006          ADD     #6,R0                   ;POINT AT NXT TBL ENTRY
4794  013600' 000767                  BR      10$                     ;GO CK IT
4795  013602' 012022          20$:    MOV     (R0)+,(R2)+             ;STORE DEV NAME
4796  013604' 011012                  MOV     (R0),(R2)
4797  013606' 000205                  RTS     R5                      ;EXIT IN-LINE
4798
4799
4800                                    ;***************
4801                                    ;
4802                                    ;  USING THE CURRENT CONTROL ROUTINE TABLE ENTRY ADDRESS
4803                                    ;  IN R4, THIS S/R WILL COMPUTE THE ENTRY'S EQUIVALENT
4804                                    ;  PROGRAM NUMBER IN ASCII.
4805                                    ;
4806                                    ;  LINKAGE;     JSR     PC,CPNASC       SUBROUTINE CALL
4807                                    ;
4808                                    ;  ENTRY:       R4 CONTAINS ADR OF CURR CONTROL ROUT TBL ENTRY
4809                                    ;
4810                                    ;  EXIT:        R1 CONTAINS EQUIVALENT PROG # IN ASCII
4811                                    ;
4812                                    ;  DESTROYS REGISTERS: R0,R1,R2
4813                                    ;
4814                                    ;***************
4815
4816  013610' 010701          CPNASC: MOV     PC,R1                   ;SET UP CTRL ROUT TBL START ADR
4817  013612' 062701  020402          ADD     #CRTBL-2-.,R1
4818  013616' 010400                  MOV     R4,R0                   ;COMPUTE ENTRY'S NUMBER
4819  013620' 160100                  SUB     R1,R0
4820  013622' 006000                  ROR     R0
4821  013624' 004567  025334  CPNAS1: JSR     R5,BTASLB               ;CONVERT # TO DECIMAL ASCII
4822  013630' 165760                  .WORD   CWA-.
4823  013632' 016701  165756          MOV     CWA+4,R1                ;GET LAST TWO DIGITS
4824  013636' 000207                  RTS     PC                      ;EXIT IN-LINE
4825                  001             .IF DF MM
4826                                  .PAGE
4827                                  ;***************
```

```
4828
4829                                    ;  THIS S/R TAKES THE HIGH ORDER 16 BITS OF AN ADR
4830                                    ;  STORED AT THE SUPPLIED LABEL AND CONVERTS IT TO A 22 BIT
4831                                    ;  ADR. THE LOW 16 BITS WILL BE IN R0 AND THE HIGH 6 BITS
4832                                    ;  IN R2. THE LOW 6 BITS OF THE 22 BIT ADR WILL BE 0'S.
4833                                    ;
4834                                    ;  LINKAGE:    JSR     R5,G22BAD       S/R CALL
4835                                    ;               .WORD   LABEL-.         REL ADR OF HIGH 16 BITS
4836                                    ;
4837                                    ;  DESTROYS R0,R1,R2
4838                                    ;
4839                                    ;***************
4840
4841                            G22BAD: MOV     R5,R1                   ;MAKE REL ADR ABSOLUTE
4842                                    ADD     (R5)+,R1
4843                                    MOV     (R1),R0                 ;GET HIGH 16 BITS
4844                                    CLR     R2                      ;RESET REG FOR HIGH 6 BITS
4845                                    MOV     #6,R1                   ;SET UP LOOP CNT
4846                            10$:    ASL     R0                      ;SHIFT BITS ONE TO THE LEFT
4847                                    ROL     R2
4848                                    DEC     R1                      ;SHIFTED 6 BITS?
4849                                    BNE     10$                     ;Y,N-10$
4850                                    RTS     R5                      ;EXIT IN-LINE
4851
4852
4853                                    ;***************
4854                                    ;
4855                                    ;  THE 22 BIT ADDRESS IN R0 & R2 WILL BE CONVERTED TO A
4856                                    ;  VIRTUAL ADDRESS BASED UPON PAR 6 WITH THE CORRECT
4857                                    ;  DISPLACEMENT. THIS ADDRESS WILL BE RETURNED IN R1.
4858                                    ;  ALSO, KERNEL'S PAR 6 WILL BE INITIALIZED TO THE CORRECT
4859                                    ;  BASE ADDRESS.
4860                                    ;
4861                                    ;  LINKAGE:     JSR     PC,CVTVAD       S/R CALL
4862                                    ;
4863                                    ;  DESTROYS R1 AND KPAR6
4864                                    ;
4865                                    ;***************
4866
4867                            CVTVAD: JSR     R4,SREG                 ;SAVE REG'S R0 - R4
4868                                    MOV     R0,R1                   ;EXTRACT LOW 6 BITS
4869                                    BIC     #177700,R1
4870                                    BIS     #P6CONS,R1              ;BASE IT UPON PAR 6
4871                                    MOV     #6,R3                   ;SET UP LOOP CNT
4872                            10$:    ASR     R2                      ;SHIFT ADR RIGHT 1 BIT
4873                                    ROR     R0
4874                                    DEC     R3                      ;DONE 6 BITS?
4875                                    BNE     10$                     ;Y,N-10$
4876                                    MOV     R0,@#KPAR6              ;STORE HIGH 16 BITS IN KPAR 6
4877                                    MOV     R1,2(SP)                ;STORE VIRTUAL ADR IN R1 ON THE STK
4878                                    JSR     R4,RREG                 ;RESTORE REGISTERS
4879                                    RTS     PC                      ;EXIT IN-LINE
4880                                    .PAGE
4381                                    ;***************
4882                                    ;
4883                                    ;  THIS SUBROUTINE WILL RESET THE UNIBUS MAP USAGE
```

```
                        ;  BITS IN THE "MAPMAP" WORDS FOR THE CURRENT PROGRAM.
                        ;  IF THE CURRENT PROGRAM DOES NOT USE THE UNIBUS MAP,
                        ;  NO ACTIONS WILL TAKE PLACE.
                        ;
                        ;  LINKAGE:     JSR     PC,MAPHKP       S/R CALL
                        ;
                        ;               R3 = PROG TABLE ADR
                        ;               R4 = CONTROL ROUT TBL ENTRY ADR
                        ;
                        ;  DESTROYS R0,R1,R2
                        ;
                        ;***************

                MAPHKP: TST     PUBMAP(R3)              ;THIS PROG USE THE UNIBUS MAP?
                        BEQ     30$                     ;Y,N-30$
                        MOVB    PUBMAP(R3),R0           ;GET 1ST MAP REG #
                        MOVB    PUBMAP+1(R3),R2         ;GET # OF REG BITS
                        CLR     R1
                        SUB     #6,R0                   ;CHG REG # INTO SHIFT CNT
                10$:    DEC     R0                      ;DONE ALL SHIFTS?
                        BEQ     20$                     ;N,Y-20$
                        ASL     R2                      ;SHIFT REG BITS ONE POSITION
                        ROL     R1
                        BR      10$                     ;GO CK SHIFT CNT NOW
                20$:    BIC     R1,MAPMAP               ;RESET THIS PROG'S UNIBUS
                        BIC     R2,MAPMAP+2             ;MAP USAGE BITS
                30$:    RTS     PC                      ;EXIT IN-LINE
                        .PAGE
                        ;***************
                        ;
                        ;  THIS SUBROUTINE CHECKS THAT A PROGRAM'S I/O AREA
                        ;  (PROG TBL + DEVICE ROUTINE + RDIO + WRIO) DOES NOT
                        ;  EXCEED 18 BIT ADDRESSING. THIS CHECK IS MADE IF 22 BIT
                        ;  ADDRESSING IS BEING USED AND THE UNIBUS MAP IS NOT.
                        ;  ALSO, IT IS MADE ONLY FOR PROGRAMS THAT WOULD NORMALLY
                        ;  REQUIRE THE UNIBUS MAP. IF IT FAILS THE TEST, A
                        ;  MESSAGE WILL BE ISSUED TO WARN THE USER THAT HIS PROGRAM
                        ;  MIGHT NOT WORK.
                        ;
                        ;  LINKAGE:     JSR     PC,CKUBAD       S/R CALL
                        ;
                        ;               KPAR4 = PROG'S HI 16 BIT BASE ADR
                        ;               R3 = PROG TBL ADR
                        ;
                        ;  DESTROYS R0,R1,R2
                        ;
                        ;***************

                CKUBAD: BIT     #BITS22,CSYSFW          ;USING 22 BIT ADDRESSING?
                        BEQ     CKUBAX                  ;Y,N-CKUBAX
                        MOV     PSRCST(R3),R0           ;GET SIZE OF DEV ROUT+RDIO+WRIO
                        BIC     #P4CONS,R0              ;IN 32 WD BLKS
                        MOV     #6,R1
                10$:    ASR     R0
                        DEC     R1
```

```
4940                                    BNE     10$
4941                                    INC     R0
4942                                    ADD     @#KPAR4,R0              ;ADD IN BASE ADR
4943                                    CMP     R0,#10000               ;PROG I/O AREA > 18 BIT ADR?
4944                                    BLOS    CKUBAX                  ;Y,N-CKUBAX
4945                                    JSR     R5,CTYPE                ;ISSUE WARNING MSG THAT PROG
4946                                    .WORD   400                     ;MAY NOT RUN
4947                                    .WORD   CERM42-.
4948                            CKUBAX: RTS     PC                      ;EXIT IN-LINE
4949                                    .PAGE
4950
4951
4952                                    ;***************
4953                                    ;
4954                                    ;  THIS ROUTINE IS A HANDLER FOR 'TRAP' INSTRUCTIONS USED
4955                                    ;  BY THE MEMORY MANAGEMENT VERSION OF MPG AND IS ENTERED
4956                                    ;  FROM THE TRAP VECTOR AT LOCATION 34. MPG USES TRAPS
4957                                    ;  WHENEVER IT IS IN USER MODE AND DESIRES TO GO TO KERNEL
4958                                    ;  MODE. THE TRAP NUMBER DETERMINES WHICH POINT IT WILL
4959                                    ;  BRANCH TO WITHIN MPG. INVALID TRAPS WILL RESULT IN AN
4960                                    ;  ERROR REPORT.
4961                                    ;
4962                                    ;***************
4963
4964                            TRP34:  MOV     R0,-(SP)                ;SAVE R0
4965                                    MOV     2(SP),R0                ;GET ADR OF TRAP INST + 2
4966                                    MFPI    -2(R0)                  ;GET TRAP INST ON STACK
4967                                    MOVB    (SP)+,R0                ;GET TRAP I.D. BYTE
4968                                    BEQ     TRAPER                  ;IS IT 0? (N,Y-TRAPER)
4969                                    BIT     #1,R0                   ;IS IT ODD?
4970                                    BNE     TRAPER                  ;N,Y-TRAPER
4971                                    CMP     R0,#MAXTRP              ;IS IT A VALID TRAP?
4972                                    BHI     TRAPER                  ;Y,N-TRAPER
4973                                    ADD     PC,R0                   ;INDEX INTO TRAP ADR TBL
4974                                    ADD     #TRPTBL-2-.,R0
4975                                    ADD     (R0),R0                 ;MAKE TRAP ADR ABS
4976                                    RTS     R0                      ;GO TO THIS TRAP'S POINT
4977
4978                            TRAPER: MOV     (SP)+,R0                ;RESTORE R0
4979                                    BR      TRP341                  ;GO TO INVALID TRAP ROUT
4980
4981                            TRPTBL: .WORD   UPGRET-.                ; 2 - USER PROGRAM RETURN TRAP
4982                                    .WORD   KSETVC-.                ; 4 - SET INTERRUPT VECTOR TRAP
4983                                    .WORD   KCLRVC-.                ; 6 - CLEAR INTERRUPT VECTOR TRAP
4984                                    .WORD   UPRTKM-.                ;10 - USER PROGRAM PRINT TRAP
4985
4986                            MAXTRP= 10
4987                                    .PAGE
4988                                    ;***************
4989                                    ;
4990                                    ;  THIS ROUTINE PROCESSES ALL INVALID TRAPS WHICH OCCUR ON
4991                                    ;  VECTORS 4, 10, 34, 114, AND 250. REGISTERS R0 THRU R7, THE PSW,
4992                                    ;  MMR0 THRU MMR3, AND THE PAR/PDR REGISTERS OF THE FAILING
4993                                    ;  STATE WILL BE DISPLAYED. IF THE TRAP OCCURRED DURING A USER
4994                                    ;  PROGRAM, AN ATTEMPT WILL BE MADE TO KILL IT AND KEEP
4995                                    ;  RUNNING. IF IT OCCURRED DURING MPG'S OPERATIONS, A HALT
```

```
4996                                    ;  WILL RESULT.
4997                                    ;
4998                                    ;***************
4999
5000                            TRP04:  MOV     #4,-(SP)        ;SET UP TRAP 04 ADR
5001                                    BR      TRPCOM          ;GO TO COMMON CODE
5002
5003                            TRP10:  MOV     #10,-(SP)       ;SET UP TRAP 10 ADR
5004                                    BR TRPCOM               ;GO TO COMMON CODE
5005
5006                            TRP34I: MOV     #34,-(SP)       ;SET UP TRAP 34 ADR
5007                                    BR      TRPCOM          ;GO TO COMMON CODE
5008
5009                            TRP114: MOV     #114,-(SP)      ;SET UP TRAP 114 ADR
5010                                    BR      TRPCOM          ;GO TO COMMON CODE
5011
5012                            TRP250: MOV     #250,-(SP)      ;SET UP TRAP 250 ADR
5013
5014                            TRPCOM: MOV     R5,-(SP)        ;SAVE REGS R0 THRU R5
5015                                    JSR     R4,SREG
5016                                    MOV     12.(SP),R0      ;GET TRAP'S ADR
5017                                    JSR     R5,BINASC       ;CONVERT TO ASCII & PUT
5018                                    .WORD   TRPADR-.        ;IN THE MSG
5019                                    CLR     TRPSTK+2        ;SET STK DISPL TO 0
5020                                    MOV     #8.,R3          ;SET UP # OF WORDS ON THE STK
5021                                    MOV     PC,R1           ;GET ADR IN MSG WHERE ASCII
5022                                    ADD     #CMMPRD-.,R1    ;DATA GOES
5023                            TRPSTK: MOV     0(SP),R0        ;GET WORD FROM STK
5024                                    JSR     R5,BINAS1       ;CONVERT IT TO ASCII & PUT IN MSG
5025                                    INC     R1              ;POINT TO NXT FLD IN MSG
5026                                    ADD     #2,TRPSTK+2     ;POINT AT NEXT WD ON STK
5027                                    DEC     R3              ;DONE ALL WDS ON STK?
5028                                    BNE     TRPSTK          ;Y,N-TRPSTK
5029                                    MOV     SP,R4           ;CALC STK PNTR AT TIME
5030                                    ADD     #16.,R4         ;OF THE TRAP
5031                                    MOV     (R4)+,R0        ;GET PSW
5032                                    MOV     R0,R5           ;SAVE IT
5033                                    JSR     R5,BINASC       ;CONVERT PSW TO ASCII & PUT
5034                                    .WORD   TRPPSW-.        ;IN THE MSG
5035                                    CLRB    TRP40M          ;INITIALIZE MSG TO AN 11/40
5036                                    MOV     @#MMR0,R0       ;GET MMR0 CONTENTS
5037                                    JSR     R5,BINASC       ;CONVERT & PUT IN THE MSG
5038                                    .WORD   TRPMR0-.
5039                                    MOV     @#MMR2,R0       ;GET MMR2 CONTENTS
5040                                    JSR     R5,BINASC       ;CONVERT & PUT IN THE MSG
5041                                    .WORD   TRPMR2-.
5042                                    BIT     #CPU40,CSYSFW   ;IS THIS AN 11/40?
5043                                    BNE     10$             ;N,Y-10$
5044                                    MOVB    #40,TRP40M      ;SET MSG FOR OTHER THAN AN 11/40
5045                                    MOV     @#MMR1,R0       ;GET MMR1 CONTENTS
5046                                    JSR     R5,BINASC       ;CONVERT & PUT IN THE MSG
5047                                    .WORD   TRPMR1-.
5048                                    MOV     @#MMR3,R0       ;GET MMR3 CONTENTS
5049                                    JSR     R5,BINASC       ;CONVERT & PUT IN THE MSG
5050                                    .WORD   TRPMR3-.
5051                            10$:    BIT     #140000,R5      ;IN USER MODE AT TIME OF TRAP?
```

```
5052                            BEQ     20$                     ;Y,N-20$
5053                            MFPI    SP                      ;GET USER MODE STK PNTR
5054                            MOV     (SP)+,R0
5055                            BR      30$                     ;BYPASS KERNEL MODE PROC
5056                    20$:    MOV     R4,R0                   ;SET UP KERNEL STK PNTR
5057                    30$:    JSR     R5,BINASC               ;CONVERT STK PNTR & PUT
5058                            .WORD   TRPSP-.                 ;IN THE MSG
5059                            JSR     R5,CTYPE                ;ISSUE MSGS WITH VALUES OF
5060                            .WORD   420                     ;ALL REGISTERS
5061                            .WORD   TRPMSG-.
5062                            .WORD   57.
5063                            JSR     R5,CTYPE
5064                            .WORD   400
5065                            .WORD   CMMPAR+7-.
5066                            JSR     R5,CTYPE
5067                            .WORD   400
5068                            .WORD   TRPPSM-.
5069                            JSR     R5,CTYPE                ;ISSUE FAILING STATE MSG
5070                            .WORD   400
5071                            .WORD   TRPFSM-.
5072                            MOV     #UPAR0,R3               ;SET UP USER MODE PAR &
5073                            MOV     #UPDR0,R5               ;PDR ADR'S
5074                            BIT     #140000,16.(SP)         ;WERE WE IN USER MODE?
5075                            BNE     35$                     ;N,Y-35$
5076                            MOV     #KPAR0,R3               ;SET UP KERNEL'S PAR &
5077                            MOV     #KPDR0,R5               ;PDR ADR'S
5078                    35$:    JSR     PC,PRTPAD               ;DISPLAY PAR & PDR REGS
5079                            BIT     #CUSPGR,CSYSFW          ;IS A USER PROG RUNNING?
5080                            BNE     40$                     ;N,Y-40$
5081                            JMP     MPGTRP                  ;GO TO MPG TRAP PROCESSING
5082                    40$:    MOV     CPGCPT,R5               ;INITIALIZE USER PROG ADR
5083                            BIT     #340,16.(SP)            ;WAS INT BEING SERVICED?
5084                            BEQ     41$                     ;Y,N-41$
5085                            JSR     R5,CTYPE                ;ISSUE 'INT BEING SERVICED'
5086                            .WORD   400                     ;MSG
5087                            .WORD   TRPISV-.
5088                            MOV     @#KPAR5,R5              ;GET PROG ADR FRM INT SRV BASE PAGE
5089                            BR      43$                     ;BYPASS OTHER TEST
5090                    41$:    TST     CPGCPT                  ;USER PROG IN CONTROL?
5091                            BEQ     MPGTRP                  ;Y,N-MPGTRP
5092                    43$:    TST     CSYSFW                  ;CONS REQ BEING PROCESSED?
5093                            BMI     MPGTRP                  ;N,Y-MPGTRP
5094                            MOV     @#KPAR4,-(SP)           ;SAVE KERNEL'S PAR 4
5095                            MOV     R5,@#KPAR4              ;SET UP PROG'S BASE ADR
5096                            MOV     #P4CONS,R3              ;SET UP FLGWD ADR
5097                            MOV     PASCIN(R3),TRPPNM       ;MOVE PROG # TO PROG # MSG
5098                            BIC     #ACTIVE,(R3)            ;RESET ITS ACTIVE FLG
5099                            BIT     #SETDED,(R3)            ;DID IT SET PRT DEV DEDICATED?
5100                            BEQ     50$                     ;Y,N-50$
5101                            BIT     #CHPINT,CSYSFW          ;IS THE PRINTER BEING USED?
5102                            BEQ     45$                     ;Y,N-45$
5103                            MOV     CLISTI,R0               ;GET ADR OF PRINTER INFO
5104                            CLR     @8.(R0)                 ;RESET PRINTER'S INT ENABLE
5105                    45$:    BIC     #CTUDED+CRUDED+CHPINT,CSYSFW  ;RESET PRT DEV DED FLGS
5106                            BIC     #SETDED+CTPRIO,(R3)
5107                    50$:    MOV     R3,R0                   ;GET ADR OF DEV ROUT KILL
```

```
5108                            ADD     #PTLGTH+DKILAD,R0        ;ROUT ADR
5109                            TST     (R0)                     ;IS THERE A KILL ROUT?
5110                            BEQ     55$                      ;Y,N-55$
5111                            ADD     (R0),R0                  ;MAKE ITS ADR ABS
5112                            MOV     @#KPAR6,-(SP)            ;SAVE PAR 6
5113                            MOV     SPPAR6,@#KPAR6           ;SET UP SHARED CODE BASE ADR
5114                            JSR     R5,(R0)                  ;GO TO KILL ROUT
5115                            MOV     (SP)+,@#KPAR6            ;RESTORE PAR 6
5116                    55$:    MOV     (SP)+,@#KPAR4            ;RESTORE PAR 4
5117                            JSR     R5,CTYPE                 ;DISPLAY USER PROG #
5118                            .WORD   400
5119                            .WORD   TRPPMG-.
5120                            MOV     #-1,STKADJ               ;HSKP STACK ADJ VALUE
5121                            MOV     #CKILEX-PG6BA+P6CONS,R0  ;SET UP USR MODE ABORT ADR
5122                            BIT     #140000,16.(SP)          ;WAS USER IN KERNEL MODE?
5123                            BNE     64$                      ;Y,N-64$
5124                            BIT     #340,16.(SP)             ;SERVICING AN INTERRUPT?
5125                            BNE     58$                      ;N,Y-58$
5126                            MOV     PC,R0                    ;SET UP KERNEL MODE ABORT ADR
5127                            ADD     #CKILEK-.,R0
5128                            BR      64$                      ;GO STORE THIS ADR
5129                    58$:    MOV     SP,R1                    ;POINT TO USER WORDS - 1
5130                            ADD     #16.,R1                  ;ON THE STK
5131                            MOV     PC,R2                    ;SET UP ADR FOR CK IF USER'S
5132                            ADD     #INTRET-.,R2             ;DEV TYPE OF INT
5133                    60$:    TST     (R1)+                    ;POINT TO NXT STK WD
5134                            CMP     (R1),R2                  ;THIS USER INT RET ADR?
5135                            BEQ     62$                      ;N,Y-62$
5136                            CMP     R1,#STACK-2              ;REACHED STK UPPER LIMIT?
5137                            BLO     60$                      ;Y,N-60$
5138                            BR      66$                      ;CAN'T FIND RETURN - LEAVE RET ADR ALONE
5139                    62$:    SUB     #12.,R1                  ;ALLOW FOR R5 THRU R0
5140                            MOV     SP,R2                    ;DETERMINE IF USER WORDS ARE ON
5141                            ADD     #18.,R2                  ;THE STK
5142                            MOV     R1,R3
5143                            SUB     R2,R3
5144                            BMI     66$                      ;BR IF STK NOT AS EXPECTED
5145                            MOV     R3,STKADJ                ;STORE STK ADJ VALUE
5146                            MOV     16.(SP),-(R1)            ;MOVE PSW UP ON STK
5147                            MOV     PC,-(R1)                 ;SET UP NEW RET ADR
5148                            ADD     #UTRPRT-.,(R1)
5149                            BR      66$                      ;BYPASS ADR STORING
5150                    64$:    MOV     R0,14.(SP)               ;STORE IT IN PC ON THE STK
5151                    66$:    JSR     R4,RREG                  ;RESTORE REGS R0 THRU R5
5152                            MOV     (SP)+,R5
5153                            CMP     (SP)+,#250               ;WAS THIS A MEM MGMNT TRAP?
5154                            BNE     68$                      ;Y,N-68$
5155                            MOV     #1001,@#MMMR0            ;CLEAR MEM MGMNT ERRORS
5156                    68$:    TST     STKADJ                   ;IS STK TO BE ADJUSTED?
5157                            BMI     UTRPEX                   ;Y,N-UTRPEX
5158                            ADD     (PC)+,SP                 ;ADJ STK PNTR IF TRAP IN USR INT
5159                    STKADJ: .WORD   0
5160                    UTRPEX: RTI                              ;RETURN & TRY TO KEEP RUNNING
5161
5162                    UTRPRT: MOV     (SP)+,R5                 ;RESTORE REGS R5 THRU R0
5163                            MOV     (SP)+,R4
```

```
5164                                    MOV     (SP)+,R3
5165                                    MOV     (SP)+,R2
5166                                    MOV     (SP)+,R1
5167                                    MOV     (SP)+,R0
5168                                    JMP     RTNINT                  ;GO TO USER INT EXIT
5169
5170
5171                            MPGTRP: JSR     R5,CTYPE                ;ISSUE MPG I.D. MSG
5172                                    .WORD   400
5173                                    .WORD   TRPMPG-.
5174                                    CLR     @#MMMR0                 ;DISABLE MEM MGMNT
5175                                    HALT                            ;TRAP WHILE MPG IN CONTROL
5176                                    JMP     RST2                    ;GO TO RESTART 2
5177            000                     .ENDC
5178
5179            001                     .IF DF MM
5180                                    .IFF
5181                                    .TITLE  MAINDEC-11-DTMGA-C   MPG INTERPRETER
5182                                    .IFT
5183                                    .TITLE  MAINDEC-11-DTMMA-B   MPG INTERPRETER
5184            000                     .ENDC
```

```
                                .SBTTL  USER PROGRAM LANGUAGE PROCESSOR

                        ;*************************************************************
                        ;
                        ;       THIS SUBROUTINE PROCESSES USER ENTRIES EITHER
                        ;       FROM THE CONSOLE TERMINAL OR FROM THE PROGRAM
                        ;       SAVE DEVICE.  IT ACCEPTS THE ENTRIES ONE
                        ;       LINE AT A TIME, PACKS IT INOT A COMPRESSED
                        ;       FORM IN AN INTERNAL BUFFER, AND THEN INSERTS
                        ;       THIS PACKED STRING IN THE PROPER POSITION
                        ;       IN RELATION TO OTHER PACKED ENTIRES AS
                        ;       DETERMINED BY THE LINE NUMBER.  CHECKS ARE
                        ;       MADE TO INSURE THAT THE PACKED CODE AND THE
                        ;       EVENTUAL MACHINE CODE WILL BOTH FIT
                        ;       WITHIN THE REMAINING MEMORY SPACE.  THIS
                        ;       PROCESS CONTINUES UNTIL THE WORD 'DONE'
                        ;       IS DETECTED IN PLACE OF A LINE NUMBER.
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,ENTER (MODIFY)(FETCH(COMPIL)
                        ;
                        ;       ENTRY-  R3      POINTS TO PROGRAM TABLE
                        ;
                        ;       EXIT-   BRANCHES TO COMPILER
                        ;
                        ;*************************************************************

013640' 105267  002217          ENTER:  INCB    LENTER                  ;SET 'ENTER' FLAG
013644' 116767  002226  002214          MOVB    LASZER,LINNBR           ;SET
013652' 116767  002220  002207          MOVB    LASZER,LINNBR+1         ;SAVED
013660' 112767  000061  002202          MOVB    #61,LINNBR+2            ;LINE NUMBER
013666' 116767  002204  002175          MOVB    LASZER,LINNBR+3         ;TO 0010
013674' 105267  002202                  INCB    FSTTME                  ;SET FIRST TIME FLAG
013700' 016363  000022  000024          MOV     PSRCST(R3),POBJST(R3)   ;INIT OBJECT START
013706' 016367  000022  001130          MOV     PSRCST(R3),LPROLN
013714' 160367  001124                  SUB     R3,LPROLN               ;INIT PROG LENGTH
013720' 000402                          BR      LLPGFC
013722' 105267  002134          MODIFY: INCB    LMODFY                  ;SET 'MODIFY' FLAG
013726' 004567  001272          LLPGFC: JSR     R5,UPCI                 ;GET LINE FROM CONSOLE
013732' 000415                          BR      LLPPAK                  ;THEN GO PACK IT
013734' 105267  001074          FETCH:  INCB    LFETCH                  ;SET 'FETCH' FLAG
013740' 016363  000022  000024          MOV     PSRCST(R3),POBJST(R3)   ;INIT OBJECT START
013746' 016367  000022  001070          MOV     PSRCST(R3),LPROLN
013754' 160367  001064                  SUB     R3,LPROLN               ;INIT PROGRAM LENGTH
013760' 004567  002120                  JSR     R5,UPLIO                ;GET LINE FROM 'SAVE' DEV
013764' 000042                          .WORD   LLPTFF-.
013766' 010367  005276          LLPPAK: MOV     R3,LPKPTP               ;SAVE PROG TBL PNTR
013772' 004567  002310                  JSR     R5,PACK                 ;PACK THE ENTRY
013776' 105767  007247                  TSTB    LERROR                  ;ERROR FLAG SET?
014002' 001011                          BNE     LLPTFF                  ;YES- GO TEST FETCH FLAG
014004' 105767  007237                  TSTB    LDELET                  ;TEST DELETE FLAG
014010' 001051                          BNE     LLPDEL                  ;BRANCH IF SET
014012' 105767  007232                  TSTB    LIGNOR                  ;IGNORE FLAG SET?
014016' 001414                          BEQ     LLPCOL                  ;NO -GO CK OBJ LENGTH
014020' 105067  007224                  CLRB    LIGNOR                  ;RESET IGNORE FLAG
```

```
5242  014024' 000506                  BR      LLPGAL          ;GO GET ANOTHER LINE
5243  014026' 105767 001002  LLPTFF: TSTB    LFETCH          ;FETCH FLAG SET?
5244  014032' 001503                  BEQ     LLPGAL          ;NO-GET ANOTHER LINE
5245  014034' 004567 172164          JSR     R5,CTYPE        ;TYPE FETCH ERROR MSG
5246  014040' 000402                  .WORD   402
5247  014042' 001004                  .WORD   LFTMSG-.
5248  014044' 000167 000630          JMP     LLPIEN          ;GO INSERT END
5249  014050' 010700          LLPCOL: MOV     PC,R0
5250  014052' 062700 007212          ADD     #LPKBUF-.,R0    ;PINT R0 AT PACK BUFR
5251  014056' 004567 013350          JSR     R5,LCKOBL       ;GO CHECK OBJECT LENGTH
5252  014062' 016701 000756          MOV     LPROLN,R1       ;GET OLD PROG LENGTH
5253  014066' 060001                  ADD     R0,R1           ;ADD NEW OBJ LENGTH
5254  014070' 010702                  MOV     PC,R2
5255  014072' 062702 007172          ADD     #LPKBUF-.,R2    ;POINT R2 AT PACK BUFR
5256  014076' 111200                  MOVB    (R2),R0
5257  014100' 060001                  ADD     R0,R1           ;ADD NEW SOURCE LENGTH
5258  014102' 062701 000012          ADD     #10.,R1         ;ADD STORAGE FOR 'END'
5259  014106' 060301                  ADD     R3,R1           ;ADD PROG TBL PNTR
5260  014110' 020167 165334          CMP     R1,CVMEND       ;IS THERE ROOM
5261  014114' 101402                  BLOS    LLPTDN          ;YES-TEST DONE FLAG
5262  014116' 000167 000546          JMP     LLPWRN          ;NO-TYPE WARNING
5263  014122' 105767 007120  LLPTDN: TSTB    LDONE           ;DONE FLAG SET?
5264  014126' 001500                  BEQ     LLPTFS          ;NO-TEST IF FIRST STMT
5265  014130' 000167 000504          JMP     LLPTEN          ;YES-TEST END
5266  014134' 004567 013230  LLPDEL: JSR     R5,LFINLN       ;SEARCH FOR EXISTING LINE
5267  014140' 000130                  .WORD   LLPLNF-.
5268  014142' 010100                  MOV     R1,R0
5269  014144' 004567 013262          JSR     R5,LCKOBL       ;GET OBJ LENGTH
5270  014150' 160067 000670          SUB     R0,LPROLN       ;SUB OBJ LENGTH
5271  014154' 111100                  MOVB    (R1),R0         ;GET LENGTH OF DELETED ENTRY
5272  014156' 060100                  ADD     R1,R0           ;ADD TO ENTRY POINTER
5273  014160' 020063 000024  LLPLP1: CMP     R0,POBJST(R3)
5274  014164' 001402                  BEQ     LLPDDN
5275  014166' 012021                  MOV     (R0)+,(R1)+     ;MOVE REST DOWN ON TOP
5276  014170' 000773                  BR      LLPLP1
5277  014172' 010163 000024  LLPDDN: MOV     R1,POBJST(R3)   ;UPDATE OBJ START ADDR
5278  014176' 160100                  SUB     R1,R0           ;RECOVER ENTRY LENGTH
5279  014200' 160067 000640          SUB     R0,LPROLN       ;SUB SOURCE LENGTH
5280  014204' 010701                  MOV     PC,R1
5281  014206' 062701 020060          ADD     #LINBUF+4-.,R1
5282  014212' 010700                  MOV     PC,R0
5283  014214' 062700 000726          ADD     #LDLMSG-.,R0
5284  014220' 012702 000011          MOV     #9.,R2
5285  014224' 112021          LLPLP2: MOVB    (R0)+,(R1)+
5286  014226' 005302                  DEC     R2
5287  014230' 001375                  BNE     LLPLP2
5288  014232' 004567 171766          JSR     R5,CTYPE        ;TYPE 'NNNN DELETED'
5289  014236' 000400                  .WORD   400
5290  014240' 020022                  .WORD   LINBUF-.
5291  014242' 105767 000566  LLPGAL: TSTB    LFETCH
5292  014246' 001404                  BEQ     LLPMFC          ;FETCH FLAG SET
5293  014250' 004567 001656          JSR     R5,UPLI         ;YES-GET LINE FROM SAVE
5294  014254' 177552                  .WORD   LLPTFF-.
5295  014256' 000643                  BR      LLPPAK
5296  014260' 004567 000740  LLPMFC: JSR     R5,UPCI         ;NO-GET LINE FROM CONSOLE
5297  014264' 000167 177476  LLPPKB: JMP     LLPPAK
```

```
5298  014270' 010701                    LLPLNF: MOV     PC,R1
5299  014272' 062701  017775                    ADD     #LINBUF+5-.,R1
5300  014276' 010700                            MOV     PC,R0
5301  014300' 062700  000653                    ADD     #LNFMSG-.,R0
5302  014304' 012702  000012                    MOV     #10.,R2
5303  014310' 112021                    LLPLP3: MOVB    (R0)+,(R1)+
5304  014312' 005302                            DEC     R2
5305  014314' 001375                            BNE     LLPLP3
5306  014316' 004567  171702                    JSR     R5,CTYPE                ;TYPE 'NNNN NOT FOUND'
5307  014322' 000402                            .WORD   402
5308  014324' 017736                            .WORD   LINBUF-.
5309  014326' 000745                            BR      LLPGAL
5310  014330' 016301  000022            LLPTFS: MOV     PSRCST(R3),R1           ;TEST IF FIRST STATEMENT
5311  014334' 020163  000024            LLPLP4: CMP     R1,POBJST(R3)
5312  014340' 001401                            BEQ     LLPADL                  ;YES-GO ADD A LINE
5313  014342' 000417                            BR      LLPCNH                  ;NO CHECK IF NEXT OLD HIGHER
5314  014344' 111200                    LLPADL: MOVB    (R2),R0                 ;GET SOURCE LENGTH
5315  014346' 060063  000024                    ADD     R0,POBJST(R3)           ;ADD LENGTH TO OBJ START
5316  014352' 060067  000466                    ADD     R0,LPROLN               ;ADD TO PROG LENGTH
5317  014356' 010200                            MOV     R2,R0
5318  014360' 004567  013046                    JSR     R5,LCKOBL               ;GO CHECK OBJ LENGTH
5319  014364' 060067  000454                    ADD     R0,LPROLN               ;ADD TO PROG LENGTH
5320  014370' 111200                            MOVB    (R2),R0                 ;GET SORURCE LENGTH AGAIN
5321  014372' 112221                    LLPLP5: MOVB    (R2)+,(R1)+             ;MOVE NEW STATEMENT
5322  014374' 005300                            DEC     R0                      ;IN AS FINAL STATEMENT
5323  014376' 001375                            BNE     LLPLP5
5324  014400' 000720                    LLPGLB: BR      LLPGAL                  ;GO GET ANOTHER LINE
5325  014402' 126261  000002  000002    LLPCNH: CMPB    2(R2),2(R1)             ;CHECK NEW LN YS OLD
5326  014410' 001402                            BEQ     LLPCLS                  ;IF = CHECK LSB LIN NBR
5327  014412' 103473                            BLO     LLPINL                  ;IF < INSERT NEW LINE
5328  014414' 000405                            BR      LLPMBH                  ;MUST BE >
5329  014416' 126261  000003  000003    LLPCLS: CMPB    3(R2),3(R1)             ;CHECK LSB OF LINE NBR
5330  014424' 001404                            BEQ     LLPRPL                  ;IF = REPLACE LINE
5331  014426' 103465                            BLO     LLPINL                  ;IF < INSERT NEW LINE
5332  014430' 111100                    LLPMBH: MOVB    (R1),R0
5333  014432' 060001                            ADD     R0,R1                   ;BUMP R1 TO END OF OLD
5334  014434' 000737                            BR      LLPLP4                  ;NOW SEE IF IT FITS
5335  014436' 010100                    LLPRPL: MOV     R1,R0
5336  014440' 004567  012766                    JSR     R5,LCKOBL               ;GET OLD OBJ LENGTH
5337  014444' 160067  000374                    SUB     R0,LPROLN               ;SUB FROM PROG LENGTH
5338  014450' 010200                            MOV     R2,R0
5339  014452' 004567  012754                    JSR     R5,LCKOBL               ;GET NEW OBJ LENGTH
5340  014456' 060067  000362                    ADD     R0,LPROLN               ;ADD TO PROG LENGTH
5341  014462' 010267  000350                    MOV     R2,LLPSNP               ;SAVE NEW POINTER
5342  014466' 010167  000346                    MOV     R1,LLPSOP               ;SAVE OLD POINTER
5343  014472' 111200                            MOVB    (R2),R0                 ;GET NEW LENGTH
5344  014474' 111102                            MOVB    (R1),R2                 ;GET OLD LENGTH
5345  014476' 160200                            SUB     R2,R0                   ;SUBTRACT OLD LENGTH
5346  014500' 060067  000340                    ADD     R0,LPROLN               ;ADJUST PROG LENGTH
5347  014504' 060063  000024                    ADD     R0,POBJST(R3)           ;AND OBJECT START
5348  014510' 005700                            TST     R0                      ;IF R0 NEGATIVE
5349  014512' 100411                            BMI     LLPMDN                  ;HOLE TOO BIG
5350
5351  014514' 016302  000024            LLPMUP: MOV     POBJST(R3),R2
5352  014520' 160002                            SUB     R0,R2
5353  014522' 016300  000024                    MOV     POBJST(R3),R0           ;POINT R0 AT THE TOP
```

```
5354  014526' 014240                LLPLP6: MOV     -(R2),-(R0)             ;MOVE EVERYTHING UP
5355  014530' 020001                        CMP     R0,R1                   ;TO MAKE A HOLE
5356  014532' 001375                        BNE     LLPLP6                  ;FOR THE NEW STATEMENT
5357  014534' 000411                        BR      LLPFLI                  ;GO FILL IN THE HOLE
5358  014536' 111102                LLPMDN: MOVB    (R1),R2
5359  014540' 060201                        ADD     R2,R1                   ;R1 POINTS AT NEXT
5360  014542' 010102                        MOV     R1,R2                   ;OLD STATEMENT
5361  014544' 060002                        ADD     R0,R2                   ;R2 POINTS WHERE IT SHD BE
5362  014546' 020263 000024         LLPLP7: CMP     R2,POBJST(R3)           ;ARE WE DONE ?
5363  014552' 001402                        BEQ     LLPFLI                  ;YES - BRANCH
5364  014554' 012122                        MOV     (R1)+,(R2)+             ;MOVE EVERYTHING DOWN
5365  014556' 000773                        BR      LLPLP7                  ;TO MAKE HOLE SMALLER
5366  014560' 016702 000252         LLPFLI: MOV     LLPSNP,R2               ;RESTORE NEW AND
5367  014564' 016701 000250                 MOV     LLPSOP,R1               ;OLD POINTERS
5368  014570' 111200                        MOVB    (R2),R0                 ;GET LENGTH OF NEW ENTRY
5369  014572' 112221                LLPLP8: MOVB    (R2)+,(R1)+             ;MOVE NEW ENTRY
5370  014574' 005300                        DEC     R0                      ;INTO PREPARED HOLE
5371  014576' 001375                        BNE     LLPLP8
5372  014600' 000677                        BR      LLPGLB                  ;THEN GET ANOTHER LINE
5373  014602' 010267 000230         LLPINL: MOV     R2,LLPSNP               ;SAVE NEW POINTER
5374  014606' 010167 000226                 MOV     R1,LLPSOP               ;SAVE OLD POINTER
5375  014612' 010200                        MOV     R2,R0
5376  014614' 004567 012612                 JSR     R5,LCKOBL               ;GET OBJECT LENGTH
5377  014620' 060067 000220                 ADD     R0,LPROLN               ;UPDATE PROG LENGTH
5378  014624' 111200                        MOVB    (R2),R0                 ;GET SOURCE LENGTH
5379  014626' 060067 000212                 ADD     R0,LPROLN               ;UPDATE PROG LENGTH
5380  014632' 060063 000024                 ADD     R0,POBJST(R3)           ;UPDATE OBJ START ADDR
5381  014636' 000726                        BR      LLPMUP                  ;GO MOVE EVERYTHING UP
5382  014640' 016300 000024         LLPTEN: MOV     POBJST(R3),R0
5383  014644' 162700 000005                 SUB     #5,R0
5384  014650' 122710 000240                 CMPB    #240,(R0)               ;LAST STATEMENT AN END
5385  014654' 001440                        BEQ     LLPFEX                  ;YES-LETS GE OUT OF HERE
5386  014656' 004567 171342                 JSR     R5,CTYPE                ;NO-TELL USER
5387  014662' 000402                        .WORD   402
5388  014664' 000230                        .WORD   LNEMSG-.
5389  014666' 000644                        BR      LLPGLB                  ;THEN GET ANOTHER LINE
5390  014670' 004567 171330         LLPWRN: JSR     R5,CTYPE                ;TYPE MEM LIMIT MSG
5391  014674' 000402                        .WORD   402
5392  014676' 000267                        .WORD   LMLMSG-.
5393  014700' 016302 000024         LLPIEN: MOV     POBJST(R3),R2           ;NOW WE INSERT AN END
5394  014704' 062763 000006 000024          ADD     #6,POBJST(R3)           ;BUMP OBJECT START BY 6
5395  014712' 112722 000006                 MOVB    #6,(R2)+                ;STORE LENGTH OF 6
5396  014716' 112722 000240                 MOVB    #240,(R2)+              ;STORE FUNCT. CODE FOR END
5397  014722' 012722 114631                 MOV     #114631,(R2)+           ;STORE 9999 FOR LINE NBR
5398  014726' 005022                        CLR     (R2)+                   ;STORE 0 FOR PHYSICAL ADDR.
5399  014730' 062767 000012 000106          ADD     #10.,LPROLN             ;UP PROG LENGTH BY 10
5400  014736' 016302 000024                 MOV     POBJST(R3),R2           ;GET OBJECT START
5401  014742' 010267 164500                 MOV     R2,CVMST                ;PUT IN FREE MEM START
5402  014746' 160302                        SUB     R3,R2                   ;SUB START OF P TABLE
5403  014750' 010263 000026                 MOV     R2,PLNGTH(R3)           ;UPDATE  P LENGTH IN TBL
5404  014754' 000417                        BR      LLPRFL                  ;EXIT
5405  014756' 016302 000024         LLPFEX: MOV     POBJST(R3),R2           ;GET OBJECT START
5406  014762' 010267 164460                 MOV     R2,CVMST                ;PUT IN FREE MEM START
5407  014766' 160302                        SUB     R3,R2                   ;SUB START OF P TABLE
5408  014770' 010263 000026                 MOV     R2,PLNGTH(R3)           ;UPDATE  P LENGTH IN TBL
5409  014774' 105767 000034                 TSTB    LFETCH                  ;TEST FETCH FLAG
```

```
 5410  015000' 001405                          BEQ     LLPRFL                  ;IF RESET, CONT EXIT
 5411  015002' 004577  163040                  JSR     R5,@GET                 ;GO VALIDATE CHECKSUM
 5412  015006' 177020                          .WORD   LLPTFF-.                ;ERROR IF NO GOOD
 5413  015010' 000000                          .WORD   0
 5414  015012' 000000                          .WORD   0
 5415  015014' 105067  001043          LLPRFL: CLRB    LENTER
 5416  015020' 105067  001036                  CLRB    LMODFY                  ;CLEAR INTERNAL FLAGS
 5417  015024' 105067  000004                  CLRB    LFETCH
 5418  015030' 000167  015230          COMPIL: JMP     LCPILE                  ;EXIT
 5419  015034' 000000                  LFETCH: .WORD   0
 5420  015036' 000000                  LLPSNP: .WORD   0                       ;SAVED NEW POINTER
 5421  015040' 000000                  LLPSOP: .WORD   0                       ;SAVED OLD POINTER
 5422  015042' 000000                  LOBJLN: .WORD   0
 5423  015044' 000000                  LPROLN: .WORD   0                       ;TOTAL PROGRAM LENGTH
 5424
 5425  015046' 054523  052116  054101  LFTMSG: .ASCIZ  /SYNTAX ERROR ON SAVED PROG-END FORCED/
       015054' 042440  051122  051117
       015062' 047440  020116  040523
       015070' 042526  020104  051120
       015076' 043517  042455  042116
       015104' 043040  051117  042503
       015112' 000104
 5426  015114' 040514  052123  051440  LNEMSG: .ASCIZ  /LAST STMNT NOT AN END/
       015122' 046524  052116  047040
       015130' 052117  040440  020116
       015136' 047105  000104
 5427  015142' 042011  046105  052105  LDLMSG: .ASCIZ  <011>/DELETED/
       015150' 042105     000
 5428  015153'    116  052117  043040  LNFMSG: .ASCIZ  /NOT FOUND/
       015160' 052517  042116     000
 5429  015165'    115  046505  046040  LMLMSG: .ASCIZ  /MEM LIMIT EXCEEDED-END FORCED/
       015172' 046511  052111  042440
       015200' 041530  042505  042504
       015206' 026504  047105  020104
       015214' 047506  041522  042105
       015222'    000
 5430          015224'                          .EVEN
```

```
                                        .SBTTL  USER PROGRAM CONSOLE INTERFACE

                                ;***************************************************************
                                ;
                                ;       THIS ROUTINE CONTROLS THE CONSOLE TERMINAL
                                ;       INPUT AND OUTPUT TO PROVIDE THE NEXT
                                ;       SEQUENTIAL LINE NUMBER, OR ACCEPT THE
                                ;       ONE SUPPLIED BY THE USER.  ITS MAIN FUNCTION
                                ;       IS TO INPUT ONE LINE OF THE USERS PROGRAM.
                                ;       ONCE ENTERED, IT DOES NOT EXIT UNTIL
                                ;       A CARRIAGE RETURN IS RECEIVED FROM THE KEYBOARD.
                                ;
                                ;       CALLING SEQUENCE-       JSR     R5,UPCI
                                ;
                                ;***************************************************************

015224'                         UPCI:
015224' 012700  000110                  MOV     #72.,R0
015230' 010701                          MOV     PC,R1
015232' 062701  017030                  ADD     #LINBUF-.,R1
015236' 112721  000015          LCLBUF: MOVB    #015,(R1)+              ;CLEAR INPUT AREA
015242' 005300                          DEC     R0
015244' 001374                          BNE     LCLBUF
015246' 004567  170752          LREADY: JSR     R5,CTYPE                ;TYPE A GTR THAN SIGN
015252' 000420                          .WORD   420
015254' 000604                          .WORD   LGTRTN-.
015256' 000001                          .WORD   1
015260' 136767  000576  000606          BITB    LMODFY,LON              ;TEST MODIFY FLAG
015266' 001130                          BNE     LRD72                   ;IF SET, GO READ 72 CHARS
015270' 004577  162540                  JSR     R5,@CTRDNE              ;READ FROM TTY - NO ECHO
015274' 000604                          .WORD   LCBUF-.
015276' 000001                          .WORD   1
015300' 126767  000574  017122          CMPB    LCBUF,LCRLF             ;TEST FOR A CARRIAGE RETURN
015306' 001006                          BNE     LSPTST                  ;GO TEST FOR A SPACE
015310' 004567  170710                  JSR     R5,CTYPE                ;TYPE CARRIAGE RETURN-LF
015314' 000420                          .WORD   420
015316' 017112                          .WORD   LCRLF-.
015320' 000002                          .WORD   2
015322' 000751                          BR      LREADY                  ;GO GET NEW INPUT
015324' 126767  000550  000532  LSPTST: CMPB    LCBUF,LSPACE            ;TEST IF IT WAS A SPACE
015332' 001415                          BEQ     LINCLN                  ;IF IT WAS, INCREMENT LINE NBR
015334' 004567  170664                  JSR     R5,CTYPE                ;OTHERWISE ECHO THE CHARACTER
015340' 000420                          .WORD   420
015342' 000536                          .WORD   LCBUF-.
015344' 000001                          .WORD   1
015346' 116767  000526  016706          MOVB    LCBUF,LINBUF            ;AND MOVE IT TO THE LINE BUFFER
015354' 004577  162450                  JSR     R5,@CTRD                ;READ FROM TTY
015360' 016703                          .WORD   LINBUF+1-.
015362' 000107                          .WORD   71.
015364' 000475                          BR      LUPDLN                  ;GO UPDATE THE LINE NUMBER
015366' 105767  000510          LINCLN: TSTB    FSTTME                  ;IF FIRST TIME THRU
015372' 001043                          BNE     LINCDN                  ;DONT INCR LINE NBR
015374' 105767  003674                  TSTB    LBDLIN                  ;TEST BAD LIN NBR FLAG
```

```
5488  015400' 001003                      BNE     LDOINC                  ;IF SET, DO INCREMENT
5489  015402' 105767  005643              TSTB    LERROR                  ;IF ERROR SET
5490  015406' 001035                      BNE     LINCDN                  ;DONT INCR LINE NBR
5491  015410' 005067  003660      LDOINC: CLR     LBDLIN                  ;CLR BAD LIN NBR FLAG
5492  015414' 010700                      MOV     PC,R0                   ;GET ADDRESS OF
5493  015416' 062700  000453              ADD     #LINNBR+3-.,R0          ;LEAST SIGNIF. ASCII DIGIT
5494  015422' 116710  000450              MOVB    LASZER,(R0)
5495  015426' 005300                      DEC     R0
5496  015430' 121067  000443              CMPB    (R0),LASNIN             ;IS IT A NINE ?
5497  015434' 001021                      BNE     LINCDG                  ;NO-INCREMENT DIGIT AND EXIT
5498  015436' 116710  000434              MOVB    LASZER,(R0)             ;YES, REPLACE IT WITH A ZERO
5499  015442' 005300                      DEC     R0                      ;MOVE POINTER TO NEXT DIGIT
5500  015444' 121067  000427              CMPB    (R0),LASNIN             ;IS IT A NINE ?
5501  015450' 001013                      BNE     LINCDG                  ;NO-INCREMENT DIGIT AND EXIT
5502  015452' 116710  000420              MOVB    LASZER,(R0)             ;YES,REPLACE IT WITH A ZERO
5503  015456' 005300                      DEC     R0                      ;MOVE POINTER TO NEXT DIGIT
5504  015460' 121067  000413              CMPB    (R0),LASNIN             ;IS IT A NINE ?
5505  015464' 001005                      BNE     LINCDG                  ;NO-INCREMENT DIGIT AND EXIT
5506  015466' 116720  000405              MOVB    LASNIN,(R0)+            ;YES-MAKE NEXT LEAST A NINE
5507  015472' 116710  000401              MOVB    LASNIN,(R0)             ;AND LSD A NINE
5508  015476' 000401                      BR      LINCDN                  ;EXIT
5509  015500' 105210              LINCDG: INCB    (R0)                    ;INCREMENT DIGIT
5510  015502' 004567  170516      LINCDN: JSR     R5,CTYPE                ;TYPE OUT LINE NUMBER
5511  015506' 000420                      .WORD   420
5512  015510' 000356                      .WORD   LINNBR-.
5513  015512' 000005                      .WORD   5
5514  015514' 016767  000346  016540      MOV     LINNBR,LINBUF           ;MOVE COMPUTED LINE NUMBER
5515  015522' 016767  000342  016534      MOV     LINNBR+2,LINBUF+2
5516  015530' 116767  000336  016530      MOVB    LINNBR+4,LINBUF+4       ;TO LINE BUFFER
5517  015536' 004577  162266              JSR     R5,@CTRD                ;READ FROM TTY
5518  015542' 016525                      .WORD   LINBUF+5-.
5519  015544' 000103                      .WORD   67.
5520  015546' 000404                      BR      LUPDLN                  ;GO UPDATE LINE NUMBER
5521  015550' 004577  162254      LRD72:  JSR     R5,@CTRD                ;READ FROM TTY
5522  015554' 016506                      .WORD   LINBUF-.
5523  015556' 000110                      .WORD   72.
5524  015560' 010701              LUPDLN: MOV     PC,R1
5525  015562' 062701  016500              ADD     #LINBUF-.,R1            ;SET POINTER AT LINE START
5526  015566' 121127  000012      10$:    CMPB    (R1),#12                ;LINE FEED?
5527  015572' 001404                      BEQ     20$                     ;YES - CHANGE TO CR
5528  015574' 122127  000015              CMPB    (R1)+,#15               ;CR?
5529  015600' 001403                      BEQ     30$                     ;YES - END OF LINE FOUND
5530  015602' 000771                      BR      10$                     ;NO - CHECK NEXT BYTE
5531  015604' 112711  000015      20$:    MOVB    #15,(R1)                ;STORE CR
5532  015610' 010701              30$:    MOV     PC,R1
5533  015612' 062701  016450              ADD     #LINBUF-.,R1            ;SET POINTER AT LINE START
5534  015616' 105067  005427              CLRB    LERROR                  ;RESET ERROR FLAG
5535  015622' 105067  000254              CLRB    FSTTME                  ;CLEAR FIRST TIME FLAG
5536  015626' 012700  000003              MOV     #3,R0                   ;SET DIGIT COUNTER TO 3
5537  015632' 121167  000226      LUPDL1: CMPB    (R1),LSPACE             ;IF NOT BLANK
5538  015636' 001011                      BNE     LUPDL2                  ;CHECK FOR NUMERIC
5539  015640' 121167  000226              CMPB    (R1),LTAB               ;IF IT WAS A TAB
5540  015644' 001401                      BEQ     LCKCR                   ;CHECK FOR CARRIAGE RETN
5541  015646' 000405                      BR      LUPDL2                  ;OTHERWISE CHECK FOR NUMERIC
5542  015650' 005201              LCKCR:  INC     R1                      ;ADVANCE CHAR POINTER
5543  015652' 122711  000015              CMPB    #15,(R1)                ;IF NOT A CARRIAGE RETURN
```

```
5544  015656' 001365                      BNE     LUPDL1          ;CHECK NEXT CHARACTER
5545  015660' 000476                      BR      LUPCIX          ;OTHERWISE EXIT
5546  015562' 121167 000210       LUPDL2: CMPB    (R1),LASZER     ;IF LESS THAN ASCII ZERO
5547  015666' 103473                      BLO     LUPCIX          ;EXIT
5548  015670' 121167 000203               CMPB    (R1),LASNIN     ;IF MORE THAN ASCII NINE
5549  015674' 101070                      BHI     LUPCIX          ;EXIT
5550  015676' 005201                      INC     R1              ;ADVANCE CHAR POINTER
5551  015700' 121167 000160               CMPB    (R1),LSPACE     ;IF SPACE NEXT
5552  015704' 001410                      BEQ     LJSTFY          ;START TO JUSTIFY
5553  015706' 121167 000160               CMPB    (R1),LTAB       ;IF TAB NEXT
5554  015712' 001405                      BEQ     LJSTFY          ;START TO JUSTIFY
5555  015714' 121127 000015               CMPB    (R1),#15        ;IF CARRIAGE RETURN
5556  015720' 001402                      BEQ     LJSTFY          ;LGTRTNT TO JUSTIFY
5557  015722' 005300                      DEC     R0              ;DECREMENT DIGIT COUNT
5558  015724' 000756                      BR      LUPDL2          ;CHECK NEXT CHARACTER
5559  015726' 010701              LJSTFY: MOV     PC,R1           ;SET R1 TO ADDRESS
5560  015730' 062701 016332               ADD     #LINBUF-.,R1    ;OF LINE BUFFER
5561  015734' 020027 000000               CMP     R0,#0           ;CHECK DIGIT COUNT
5562  015740' 001440                      BEQ     LSAVLN          ;IF ZERO, DONT JUSTIFY
5563  015742' 002417                      BLT     LLJUST          ;IF NEGATIVE, LEFT JUSTIFY
5564  015744' 062701 000110               ADD     #72.,R1         ;IF POSITIVE, RIGHT JUSTIFY
5565  015750' 010102                      MOV     R1,R2           ;SET R1 & R2 TO LINBUF + 72
5566  015752' 160001                      SUB     R0,R1           ;SEPARATE BY AMMOUNT TO JUSTIFY
5567  015754' 010700                      MOV     PC,R0           ;SET R0 TO ADDRESS
5568  015756' 062700 016304               ADD     #LINBUF-.,R0    ;OF LINE BUFFER
5569  015762' 114142              RJUST1: MOVB    -(R1),-(R2)     ;START SHIFTING LINE
5570  015764' 020100                      CMP     R1,R0           ;SHIFT COMPLETE ?
5571  015766' 001375                      BNE     RJUST1          ;NO, SHIFT SOME MORE
5572  015770' 116742 000102       LRJST2: MOVB    LASZER,-(R2)    ;FILL IN LEADING ZEROES
5573  015774' 020200                      CMP     R2,R0
5574  015776' 001374                      BNE     LRJST2
5575  016000' 000420                      BR      LSAVLN          ;GO SAVE JUSTIFIED LINE NBR
5576  016002' 010102              LLJUST: MOV     R1,R2           ;SET R2 TO LINBUF ADDRESS
5577  016004' 160002                      SUB     R0,R2           ;SEPARATE BY AMMOUNT TO JUSTIFY
5578  016006' 010100                      MOV     R1,R0
5579  016010' 116720 000050       LSPFL:  MOVB    LSPACE,(R0)+    ;CLEAR LEADING ZEROES
5580  016014' 020002                      CMP     R0,R2           ;ALL CLEAR ?
5581  016016' 001374                      BNE     LSPFL           ;IF NOT, CLEAR SOME MORE
5582  016020' 111221              LLJST1: MOVB    (R2),(R1)+      ;KEEP SHIFTING
5583  016022' 116722 000036               MOVB    LSPACE,(R2)+    ;AND INSERTING
5584  016026' 121267 000032               CMPB    (R2),LSPACE     ;IF LAST CHAR NOT A LSPACE
5585  016032' 001403                      BEQ     LSAVLN          ;SHIFT SOME MORE
5586  016034' 121267 000032               CMPB    (R2),LTAB       ;IF LAST CHAR NOT A LTAB
5587  016040' 001367                      BNE     LLJST1          ;SHIFT SOME MORE
5588  016042' 016767 016214 000016 LSAVLN: MOV    LINBUF,LINNBR   ;SAVE CURRENT
5589  016050' 016767 016210 000012       MOV     LINBUF+2,LINNBR+2 ;LINNBR
5590  016056' 000205              LUPCIX: RTS     R5              ;RETURN TO CALLING PROGRAM
5591
5592  016060' 000076              LGTRTN: .ASCIZ  />/
5593  016062'    000              LMODFY: .BYTE   0
5594  016063'    000              LENTER: .BYTE   0
5595  016064' 000040              LSPACE: .ASCIZ  / /
5596  016066' 030060 030060       LINNBR: .ASCII  /0000/          ;LINE NUMBER
5597  016072' 000011              LTAB:   .ASCIZ  <11>            ;HORIZLONTAL LTAB
5598  016074'    001              LON:    .BYTE   1
5599  016075'    000              LOFF:   .BYTE   0
```

```
5600  016076'    060          LASZER: .ASCII  /0/          ;ASCII ZERO
5601  016077'    071          LASNIN: .ASCII  /9/          ;ASCII NINE
5602  016100'    000          LCBUF:  .BYTE   0            ;CHARACTER BUFFER
5603  016101'    000                  .BYTE   0
5604  016102' 000000          FSTTME: .WORD   0            ;FIRST TIME FLAG
```

```
                                .SBTTL  USER PROGRAM LIBRARY INTERFACE

                        ;*******************************************************************
                        ;
                        ;       THIS ROUTINE READS PROGRAM SOURCE STATEMENTS
                        ;       IN BLOCKS FROM THE LIBRARY DEVICE
                        ;       AND TRANSFERS ONLY ONE STATEMENT TO THE
                        ;       LINE BUFFER EACH TIME IT IS CALLED.
                        ;
                        ;       CALLING SEQUENCE
                        ;
                        ;               JSR     R5,UPLIO                FIRST CALL
                        ;               .WORD   ERRTN-.
                        ;
                        ;               JSR     R5,UPLI                 SUBSEQUENT CALLS
                        ;               .WORD   ERRTN-.
                        ;
                        ;*******************************************************************

016104' 004577  161736          UPLIO:  JSR     R5,@GET                 ;GET FIRST BLOCK
016110' 000130                          .WORD   LLIERT-.
016112' 000000          LLIEW1: .WORD   0                               ;ADDRESS SUPPLIED  BY EXEC
016114' 000000          LLIEW2: .WORD   0                               ;BYTE COUNT ALSO SUPPLIED
016116' 016767  177770  000160          MOV     LLIEW1,LLIBPT           ;UPDATE BLOCK POINTER
016124' 016767  177764  000150          MOV     LLIEW2,LLIBCT           ;UPDATE DATA LENGTH
016132' 010702                  UPLI:   MOV     PC,R2
016134' 062702  016126                  ADD     #LINBUF-.,R2            ;SET LINE POINTER
016140' 016704  000140                  MOV     LLIBPT,R4               ;GET BLOCK POINTER
016144' 005767  000132          LLITBC: TST     LLIBCT                  ;BYTES LEFT IN BLOCK?
016150' 001410                          BEQ     LLITMB                  ;NO - TEST IF MORE BLOCKS
016152' 112422                          MOVB    (R4)+,(R2)+             ;MOVE CHAR TO LINE BUF
016154' 005367  000122                  DEC     LLIBCT                  ;DECREMENT DATA LENGTH
016160' 126227  177777  000012          CMPB    -1(R2),#12              ;WAS IT A LINE FD?
016166' 001430                          BEQ     LLIEXT                  ;IF YES EXIT
016170' 000765                          BR      LLITBC                  ;CHECK IF END OF BLOCK
016172' 016704  161612          LLITMB: MOV     BUF,R4
016176' 005714                          TST     (R4)                    ;MORE BLOCKS LEFT?
016200' 001413                          BEQ     LLIERR                  ;ERROR IF NOT
016202' 004577  161640                  JSR     R5,@GET                 ;GET NEXT BLOCK
016206' 000032                          .WORD   LLIERT-.
016210' 000000          LLIEW3: .WORD   0                               ;ADDRESS SUPPLIED BY EXEC
016212' 000000          LLIEW4: .WORD   0                               ;BYTE  COUNT ALSO SUPPLIED
016214' 016704  177770                  MOV     LLIEW3,R4               ;POINT R4 AT DATA
016220' 016767  177766  000054          MOV     LLIEW4,LLIBCT           ;UPDATE DATA  LENGTH
016226' 000746                          BR      LLITBC
016230' 004567  167770          LLIERR: JSR     R5,CTYPE                ;TYPE FETCH ERROR MSG
016234' 000412                          .WORD   412
016236' 000022                          .WORD   LLIEM1-.
016240' 105267  005005          LLIERT: INCB    LERROR                  ;SET ERROR FLAG
016244' 061505                          ADD     (R5),R5                 ;CALC ERROR RETURN
016246' 000205                          RTS     R5                      ;AND TAKE IT
016250' 010467  000030          LLIEXT: MOV     R4,LLIBPT               ;SAVE BLOCK BUF PNTR
016254' 005725                          TST     (R5)+                   ;BYPASS ERR RETURN ADR
016256' 000205                          RTS     R5                      ;RETURN
```

```
 5662  016260' 046506  020124  047117  LLIEM1: .ASCIZ  /FMT ON SAVED FILE/
       016266' 051440  053101  042105
       016274' 043040  046111  000105
 5663                          .EVEN
 5664  016302' 000000          LLIBCT: .WORD   0                       ;DATA LENGTH
 5665  016304' 000000          LLIBPT: .WORD   0                       ;BLOCK CHAR POINTER
```

```
                        .SBTTL  PACK SUBROUTINE

                ;*****************************************************************
                ;
                ;       THIS SUBROUTINE EXAMINES ONE LINE OF A
                ;       USERS PROGRAM ONE WORD AT AT TIME AND
                ;       CONVERTS IT TO AN INTERNAL PACKED CODE.
                ;
                ;       CALLING SEQUENCE        JSR     R5,PACK
                ;
                ;       ENTRY   USERS STATEMENT IS IN LINBUF
                ;
                ;       EXIT    PACKED STATEMENT IS IN LPKBUF
                ;
                ;*****************************************************************

016306' 105067 004737   PACK:   CLRB    LERROR                  ;RESET ERROR FLAG
016312' 005067 004730           CLR     LDONE                   ;RESET DONE AND DELETE FLAGS
016316' 010346                  MOV     R3,-(SP)
016320' 010446                  MOV     R4,-(SP)                ;SAVE R3 AND R4
016322' 012700 000110           MOV     #72.,R0
016326' 010703                  MOV     PC,R3
016330' 062703 004734           ADD     #LPKBUF-.,R3
016334' 105023          LPKCBF: CLRB    (R3)+                   ;CLEAR PACK BUFFER
016336' 005300                  DEC     R0
016340' 001375                  BNE     LPKCBF
016342' 162703 000110           SUB     #72.,R3                 ;SET OUTPUT POINTER
016346' 010702                  MOV     PC,R2
016350' 062702 015711           ADD     #LINBUF-1-.,R2          ;SET INPUT POINTER
016354' 010267 010244           MOV     R2,LSCNPT               ;AND SAVE IT
016360' 004567 010022           JSR     R5,SCAN                 ;SCAN FIRST WORD
016364' 010702                  MOV     PC,R2
016366' 062702 004664           ADD     #LASDON-.,R2
016372' 012704 000003           MOV     #3,R4
016376' 160400                  SUB     R4,R0
016400' 122220          LPKLP1: CMPB    (R2)+,(R0)+             ;WAS FIRST WORD 'DONE'
016402' 001025                  BNE     LPKCLN                  ;NO-GO CONVERT LINE NBR
016404' 005304                  DEC     R4
016406' 001374                  BNE     LPKLP1
016410' 005267 004632   LPKSDN: INC     LDONE                   ;YES-SET DONE FLAG
016414' 000416                  BR      LGOEXT                  ;THEN FIND EXIT
016416' 105267 004626   LPKSIG: INCB    LIGNOR                  ;OTHERWISE SET IGNOR
016422' 000413                  BR      LGOEXT                  ;AND EXIT
016424' 105267 004617   LPKSDL: INCB    LDELET                  ;SET DELETE FLAG
016430' 000410                  BR      LGOEXT                  ;AND EXIT
016432' 004567 167566   LPKNLN: JSR     R5,CTYPE
016436' 000412                  .WORD   412
016440' 170357                  .WORD   CERM16-.
016442' 105267 002626           INCB    LBDLIN                  ;SET BAD LINE NBR FLAG
016446' 105267 004577           INCB    LERROR                  ;SET ERROR FLAG
016452' 000167 000756   LGOEXT: JMP     LPKEXT
016456' 005304          LPKCLN: DEC     R4
016460' 060400                  ADD     R4,R0
016462' 004567 023662           JSR     R5,ASDECO               ;CONVERT LINE NUMBER
016466' 177744                  .WORD   LPKNLN-.                ;TO PACKED DECIMAL
```

```
5723  016470' 116063 000001 000002          MOVB    1(R0),2(R3)
5724  016476' 116063 000002 000003          MOVB    2(R0),3(R3)          ;STORE IN PACK BUFFER
5725  016504' 004567 007676                 JSR     R5,SCAN              ;GET NEXT WORD
5726  016510' 122710 000015                 CMPB    #15,(R0)             ;IF CARRIAGE RETURN
5727  016514' 001740                        BEQ     LPKSIG               ;GO SET IGNORE FLAG
5728  016516' 010702                LPKILU: MOV     PC,R2
5729  016520' 062702 003230                 ADD     #LPKTBE-.,R2
5730  016524' 010212                        MOV     R2,(R2)
5731  016526' 010702                        MOV     PC,R2
5732  016530' 062702 002560                 ADD     #LPKTBL+6-.,R2       ;SET POINTER TO PACK TABLE
5733  016534' 005267 002536                 INC     LPKIFL               ;SET INTERNAL TBL FLAG
5734  016540' 005200                LPKL25: INC     R0
5735  016542' 026702 003202        LPKCTE: CMP     LPKTBE,R2            ;ARE WE AT TBL END?
5736  016546' 001034                        BNE     LPKSCC               ;NO-SAVE CHAR COUNT
5737  016550' 005767 002522                 TST     LPKIFL               ;IF INTERNAL FLAG CLR
5738  016554' 001401                        BEQ     LPKEB1               ;TAKE ERROR BRANCH
5739  016556' 000402                        BR      LPKSEX               ;SET UP FOR EXTERNAL TBL
5740  016560' 000167 000644        LPKEB1: JMP     LPKERR               ;ELSE TAKE ERROR EXIT
5741  016564' 005067 002506        LPKSEX: CLR     LPKIFL               ;CLEAR INTERNAL TBL FLAG
5742  016570' 016702 002474                 MOV     LPKPTP,R2            ;DETERMINE END
5743  016574' 062702 000352                 ADD     #PTLGTH+DVVTEP,R2    ;OF
5744  016600' 061202                        ADD     (R2),R2              ;EXTERNAL PACK TABLE
5745  016602' 062702 000006                 ADD     #6,R2
5746  016606' 010267 003136                 MOV     R2,LPKTBE
5747  016612' 016702 002452                 MOV     LPKPTP,R2            ;POINT R2
5748  016616' 062702 000350                 ADD     #PTLGTH+DVPTEP,R2    ;AT START OF
5749  016622' 061202                        ADD     (R2),R2              ;EXT PACK TABLE
5750  016624' 026702 003120                 CMP     LPKTBE,R2            ;ARE WE AT TBL END?
5751  016630' 001753                        BEQ     LPKEB1               ;YES - ERROR
5752  016632' 062702 000006                 ADD     #6,R2                ;ADJUST R2
5753  016636' 000741                        BR      LPKCTE               ;CHECK IF START = END
5754  016640' 010104                LPKSCC: MOV     R1,R4                ;SAVE CHAR COUNT
5755  016642' 010267 004412                 MOV     R2,LPKPSV            ;AND TABLE POINTER
5756  016646' 124042                LPKLP3: CMPB    -(R0),-(R2)          ;COMPARE ONE CHAR
5757  016650' 001011                        BNE     LPKNXI               ;IF NOT=CHECK NEXT INST.
5758  016652' 005304                        DEC     R4                   ;DECREMENT CHAR COUNT
5759  016654' 001374                        BNE     LPKLP3               ;LOOP IF MORE TO GO
5760  016656' 022701 000006                 CMP     #6,R1                ;IF WORD COUNT = 6
5761  016662' 001414                        BEQ     LPKIFD               ;INSTRUCTION FOUND
5762  016664' 122742 000040                 CMPB    #40,-(R2)            ;IF NO LEADING BL IN TBL
5763  016670' 001366                        BNE     LPKLP3               ;CHECK NEXT INST
5764  016672' 000410                        BR      LPKIFD               ;INSTRUCTION FOUND!
5765  016674' 062767 000010 004356 LPKNXI: ADD     #8.,LPKPSV           ;BUMP POINTER
5766  016702' 016702 004352                 MOV     LPKPSV,R2            ;PUT IT IN R2
5767  016706' 016700 007712                 MOV     LSCNPT,R0
5768  016712' 000712                        BR      LPKL25               ;CHECK NEXT TABLE ENTRY
5769  016714' 016702 004340        LPKIFD: MOV     LPKPSV,R2
5770  016720' 121227 000257                 CMPB    (R2),#257            ;WAS IT DELETE ?
5771  016724' 001637                        BEQ     LPKSDL               ;YES - GO SET FLAG
5772  016726' 121227 000256                 CMPB    (R2),#256            ;WAS IT DONE ?
5773  016732' 001626                        BEQ     LPKSDN               ;YES-SET DONE FLAG
5774  016734' 111263 000001                 MOVB    (R2),1(R3)           ;STORE BASIC FUNCTION CODE
5775  016740' 105762 000001                 TSTB    1(R2)                ;CHECK FLAG BYTE
5776  016744' 001402                        BEQ     LPKCIO
5777  016746' 000167 001104                 JMP     LPKSPH               ;SET MEANS SPECIAL HANDLING
5778  016752' 121227 000317        LPKCIO: CMPB    (R2),#317
```

```
5779  016756' 101014                        BHI     LPKCDE          ;BRANCH IF I/O CMD
5780  016760' 121227  000120                CMPB    (R2),#120
5781  016764' 103417                        BLO     LPKSSP          ;BRANCH IF NOT READ
5782  016766' 121227  000137                CMPB    (R2),#137
5783  016772' 101406                        BLOS    LPKCDE          ;WRITE
5784  016774' 121227  000160                CMPB    (R2),#160
5785  017000' 103411                        BLO     LPKSSP          ;OR BREAK
5786  017002' 121227  000167                CMPB    (R2),#167
5787  017006' 101006                        BHI     LPKSSP
5788  017010' 111200                LPKCDE: MOVB    (R2),R0         ;FUNCTION CODE TO R0
5789  017012' 016702  002252                MOV     LPKPTP,R2       ;P TBL PNTR TO R2
5790  017016' 004567  010612                JSR     R5,FENTRY       ;CHECK IF FC SUPPORTED
5791  017022' 000352                        .WORD   LPKDER-.
5792  017024' 016767  007574  007576 LPKSSP: MOV     LSCNPT,LSPTSV   ;SAVE SCAN POINTER
5793  017032' 016702  004222                MOV     LPKPSV,R2       ;GET PACK POINTER
5794  017036' 121227  000237                CMPB    (R2),#237       ;IF NOT MULTIPLE FORMAT
5795  017042' 101045                        BHI     LPKSVP          ;DONT GO ADD WC TO FC
5796  017044' 005004                        CLR     R4              ;SET WORD COUNT=0
5797  017046' 004567  007334        LPKLP4: JSR     R5,SCAN         ;SCAN FOR NEXT WORD
5798  017052' 122710  000015                CMPB    #15,(R0)        ;WAS IT A CARRIAGE RET?
5799  017056' 001402                        BEQ     LPKCBK          ;YES-CHECK IF BREAK
5800  017060' 105204                        INCB    R4              ;NO-INCREMENT WORD COUNT
5801  017062' 000771                        BR      LPKLP4          ;SCAN NEXT WORD
5802  017064' 120427  000003        LPKCBK: CMPB    R4,#3           ;WORD COUNT < 3 ?
5803  017070' 103430                        BLO     LPKUFC          ;YES - CONTINUE
5804  017072' 126327  000001  000160        CMPB    1(R3),#160      ;IF NOT A BREAK
5805  017100' 103407                        BLO     LPKCRW          ;CHECK FOR RD OR WR
5806  017102' 126327  000001  000167        CMPB    1(R3),#167      ;IF NOT RD, WR OR BK
5807  017110' 101020                        BHI     LPKUFC          ;GO UPDATE F C
5808  017112' 112704  000003                MOVB    #3,R4           ;JAM 3 INTO WD CNT
5809  017116' 000415                        BR      LPKUFC
5810  017120' 120427  000005        LPKCRW: CMPB    R4,#5           ;WORD COUNT < 5
5811  017124' 103412                        BLO     LPKUFC          ;YES - CONTINUE
5812  017126' 126327  000001  000120        CMPB    1(R3),#120      ;IF NOT A READ
5813  017134' 103406                        BLO     LPKUFC          ;CONTINUE
5814  017136' 126327  000001  000137        CMPB    1(R3),#137      ;IF NOT A WRITE
5815  017144' 101002                        BHI     LPKUFC          ;CONTINUE
5816  017146' 112704  000005                MOVB    #5,R4           ;JAM 5 INTO WORD CNT
5817  017152' 150463  000001        LPKUFC: BISB    R4,1(R3)        ;ADD WORD COUNT TO F.C.
5818  017156' 010704                LPKSVP: MOV     PC,R4           ;SET POINTER TO
5819  017160' 062704  002572                ADD     #LMPNTR-.,R4    ;MODEL STATEMENT VECTOR TBL
5820  017164' 121463  000001        LPKLP5: CMPB    (R4),1(R3)      ;CHECK F.C. IN TABLE
5821  017170' 001450                        BEQ     LPKSMP          ;BRANCH IF FOUND
5822  017172' 122714  000377                CMPB    #377,(R4)       ;TEST IF END INT TBL
5823  017176' 001403                        BEQ     LPKSEV          ;YES-SET EXT VECTOR
5824  017200' 062704  000004                ADD     #4,R4           ;BUMP POINTER
5825  017204' 000767                        BR      LPKLP5          ;CHECK NEXT ENTRY
5826  017206' 016704  002056        LPKSEV: MOV     LPKPTP,R4       ;DETERMINE END
5827  017212' 062704  000354                ADD     #PTLGTH+DVCTEP,R4 ;OF
5828  017216' 061404                        ADD     (R4),R4         ;EXTERNAL VECTOR TBL
5829  017220' 010467  002054                MOV     R4,LMSTBE
5830  017224' 016704  002040                MOV     LPKPTP,R4       ;POINT R4 AT START
5831  017230' 062704  000352                ADD     #PTLGTH+DVVTEP,R4 ;OF
5832  017234' 061404                        ADD     (R4),R4         ;EXTERN. VECTOR TABLE
5833  017236' 020467  002036        LPKLP6: CMP     R4,LMSTBE       ;R4 AT TBL END?
5834  017242' 001472                        BEQ     LPKERR          ;YES - ERROR
```

```
5835  017244' 121463 000001                CMPB   (R4),1(R3)          ;CHECK F.C. IN TABLE
5836  017250' 001403                       BEQ    LPKSEP              ;BRANCH IF FOUND
5837  017252' 062704 000004                ADD    #4,R4               ;BUMP POINTER
5838  017256' 000767                       BR     LPKLP6              ;CHECK NEXT ENTRY
5839  017260' 016767 007344 007336 LPKSEP: MOV    LSPTSV,LSCNPT       ;RESTORE SCAN POINTER
5840  017266' 062704 000002                ADD    #2,R4
5841  017272' 062703 000006                ADD    #6,R3               ;ADJUST PACK POINTER
5842  017276' 016702 001766                MOV    LPKPTP,R2
5843  017302' 062702 000242                ADD    #PTLGTH,R2
5844  017306' 061402                       ADD    (R4),R2             ;SET EXTERNAL MDL PTR
5845  017310' 000411                       BR     LBASE1
5846  017312' 016767 007312 007304 LPKSMP: MOV    LSPTSV,LSCNPT       ;RESTORE SCAN POINTER
5847  017320' 062704 000002                ADD    #2,R4
5848  017324' 062703 000006                ADD    #6,R3               ;ADJUST PACK POINTER
5849  017330' 011402                       MOV    (R4),R2
5850  017332' 060702                       ADD    PC,R2               ;SET MODEL POINTER
5851  017334' 010267 003722        LBASE1: MOV    R2,LPKMPS
5852  017340' 122712 000377        LPKCNB: CMPB   #377,(R2)
5853  017344' 001503                       BEQ    LPKVAR              ;BRANCH FOR VARIABLE
5854  017346' 122712 000376                CMPB   #376,(R2)
5855  017352' 001440                       BEQ    LPKLIN              ;BRANCH FOR LINE NBR
5856  017354' 122712 000373                CMPB   #373,(R2)           ;IF END OF PRINT STATEMENT
5857  017360' 001425                       BEQ    LPKEXT              ;EXIT
5858  017362' 122712 000375                CMPB   #375,(R2)
5859  017366' 001005                       BNE    LPKCEM
5860  017370' 000167 000546                JMP    LPKBIT              ;BRANCH FOR BIT NBR
5861  017374' 105267 003651        LPKDER: INCB   LERROR              ;SET ERROR FLAG
5862  017400' 000415                       BR     LPKEXT              ;GO TO EXIT
5863  017402' 105712               LPKCEM: TSTB   (R2)                ;END OF MODEL ?
5864  017404' 001404                       BEQ    LPKCUE              ;YES-CHECK USER ENTRY
5865  017406' 005202                       INC    R2                  ;BUMP MODEL POINTER
5866  017410' 010267 003646                MOV    R2,LPKMPS           ;AND SAVE IT
5867  017414' 000747                       BR     LBASE1              ;CHECK NEXT BYTE OF MDL
5868  017416' 004567 006764        LPKCUE: JSR    R5,SCAN             ;END OF USER ENTRY?
5869  017422' 122710 000015                CMPB   #15,(R0)
5870  017426' 001402                       BEQ    LPKEXT              ;YES-EXIT
5871  017430' 004567 004664        LPKERR: JSR    R5,SYNTAX           ;NO-REPORT ERROR
5872  017434' 010702               LPKEXT: MOV    PC,R2
5873  017436' 062702 003626                ADD    #LPKBUF-.,R2
5874  017442' 160203                       SUB    R2,R3               ;CALCULATE PACKED LENGTH
5875  017444' 110312               LPKSTL: MOVB   R3,(R2)             ;STORE IN FIRST BYTE
5876  017446' 012604                       MOV    (SP)+,R4
5877  017450' 012603                       MOV    (SP)+,R3            ;RESTORE STACK
5878  017452' 000205                       RTS    R5                  ;RETURN TO CALLING PGM
5879  017454' 004567 006726        LPKLIN: JSR    R5,SCAN             ;FIND NEXT USER WORD
5880  017460' 122710 000015                CMPB   #15,(R0)            ;WAS IT CARRIAGE RET
5881  017464' 001761                       BEQ    LPKERR              ;YES MEANS ERROR
5882  017466' 112723 000114                MOVB   #'L,(R3)+           ;STORE L FOR LINE NBR
5883  017472' 004567 022652                JSR    R5,ASDECO           ;CONVERT TO
5884  017476' 177732                       .WORD  LPKERR-.            ;PACKED DECIMAL
5885  017500' 111023                       MOVB   (R0),(R3)+
5886  017502' 116023 000001                MOVB   1(R0),(R3)+         ;STORE IN PACKED CODE
5887  017506' 116023 000002                MOVB   2(R0),(R3)+
5888  017512' 105710                       TSTB   (R0)                ;ERROR IF LINE NBR
5889  017514' 001345                       BNE    LPKERR              ;GREATER THAN 4 DIGITS
5890  017516' 112763 000377 177775        MOVB   #377,-3(R3)         ;REDUCE TO 4 DIGITS
```

```
5891  017524' 005267  003532          LPKAMP: INC     LPKMPS          ;ADVANCE MODEL POINTER
5892  017530' 004567  006652                  JSR     R5,SCAN         ;SCAN CONNECTING WORD
5893  017534' 016702  003522                  MOV     LPKMPS,R2       ;RESTORE MODEL POINTER
5894  017540' 122710  000015                  CMPB    #15,(R0)        ;WAS IT A CARRIAGE RET?
5895  017544' 001273                          BNE     LBASE1          ;NO-CHECK NEXT BYTE
5896  017546' 105712                          TSTB    (R2)            ;END OF MODEL STATEMENT?
5897  017550' 001327                          BNE     LPKERR          ;NO-ERROR
5898  017552' 000730                          BR      LPKEXT          ;YES-EXIT
5899  017554' 004567  006626          LPKVAR: JSR     R5,SCAN         ;SCAN VARIABLE
5900  017560' 122710  000015                  CMPB    #15,(R0)        ;WAS IT A CARRIAGE RET?
5901  017564' 001721                          BEQ     LPKERR          ;YES-ERROR
5902  017566' 005301                          DEC     R1
5903  017570' 001425                          BEQ     LPKOCT
5904  017572' 160100                          SUB     R1,R0
5905  017574' 126067  000001  176274          CMPB    1(R0),LASZER    ;SECOND DIGIT NON NUMERIC?
5906  017602' 103457                          BLO     LPKTAG          ;YES-PACK TAG
5907  017604' 126067  000001  176265          CMPB    1(R0),LASNIN    ;SECOND DIGIT NON NUMERIC?
5908  017612' 101053                          BHI     LPKTAG          ;YES-PACK TAG
5909  017614' 122710  000104                  CMPB    #'D,(R0)        ;FIRST CHAR A 'D'
5910  017620' 001431                          BEQ     LPKDEC          ;YES-PACK DECIMAL
5911  017622' 122710  000117                  CMPB    #'O,(R0)        ;FIRST CHAR AN 'O'
5912  017626' 001406                          BEQ     LPKOCT          ;YES-PACK OCTAL
5913
5914  017630' 121067  176242                  CMPB    (R0),LASZER     ;FIRST DIGIT NON NUMERIC?
5915  017634' 103442                          BLO     LPKTAG          ;YES-PACK TAG
5916  017636' 121067  176235                  CMPB    (R0),LASNIN     ;FIRST DIGIT NON NUMERIC?
5917  017642' 101037                          BHI     LPKTAG          ;YES-PACK TAG
5918  017644' 112723  000040          LPKOCT: MOVB    #40,(R3)+       ;STORE A BLANK
5919  017650' 060100                          ADD     R1,R0           ;POINTER TO RIGHT DIGIT
5920  017652' 005201                          INC     R1              ;RESTORE CHAR COUNT
5921  017654' 004567  022462                  JSR     R5,ASDEC1       ;CONVERT TO PACKED DECIMAL
5922  017660' 177550                          .WORD   LPKERR-.        ;REPRESENTATION OF OCT
5923  017662' 111023                          MOVB    (R0),(R3)+
5924  017664' 116023  000001                  MOVB    1(R0),(R3)+     ;STORE PACKED DECIMAL (OCTAL)
5925  017670' 116023  000002                  MOVB    2(R0),(R3)+
5926  017674' 004567  022232                  JSR     R5,OCTBIN       ;CONVERT TO BINARY
5927  017700' 177530                          .WORD   LPKERR-.        ;JUST TO CHECK FOR ERROR
5928  017702' 000710                          BR      LPKAMP          ;IF OK ADVANCE MODEL PTR.
5929  017704' 112723  000104          LPKDEC: MOVB    #'D,(R3)+       ;STORE A 'D'
5930  017710' 060100                          ADD     R1,R0           ;POINTER TO RIGHT DIGIT
5931  017712' 004567  022424                  JSR     R5,ASDEC1       ;CONVERT TO PACKED DECIMAL
5932  017716' 177512                          .WORD   LPKERR-.
5933  017720' 111023                          MOVB    (R0),(R3)+
5934  017722' 116023  000001                  MOVB    1(R0),(R3)+     ;STORE PACKED DECIMAL
5935  017726' 116023  000002                  MOVB    2(R0),(R3)+
5936  017732' 004567  022160                  JSR     R5,DECBIN       ;CONVERT TO BINARY
5937  017736' 177472                          .WORD   LPKERR-.        ;JUST TO CHECK FOR ERROR
5938  017740' 000671                          BR      LPKAMP          ;IF OK ADVANCE MODEL PTR.
5939  017742' 005201          LPKTAG: INC     R1
5940  017744' 020127  000004                  CMP     R1,#4           ;CHECK IF TAG LENGTH<=4
5941  017750' 101401                          BLOS    LPKCKT
5942  017752' 000626          JLPKER: BR      LPKERR          ;ERROR IF IT ISNT
5943  017754' 010367  001312  LPKCKT: MOV     R3,LPKPKP
5944  017760' 016703  001304          MOV     LPKPTP,R3
5945  017764' 004567  006642          JSR     R5,TAGLUP       ;CHECK IF TAG VALID
5946  017770' 177440                  .WORD   LPKERR-.
```

```
 5947                                                          ;*****DQ11 SPECIAL CODE STARTS HERE
 5948                001                          .IF DF MM
 5949                                             CMP     2(R2),#2          ;THIS MAY BE "OPSW" TAG
 5950                                             BEQ     LPKTGD            ;BR IF IT COULD BE
 5951                000                          .ENDC
 5952  017772' 026227  000002  000017             CMP     2(R2),#17         ;VALUE GREATER THAN 17?
 5953  020000' 101007                             BHI     LPKTGD            ;ASSUME VALID TAG
 5954  020002' 010703                             MOV     PC,R3
 5955  020004' 062703  003261                     ADD     #LPKBUF+1-.,R3
 5956  020010' 122713  000371                     CMPB    #371,(R3)         ;SELREG COMMAND?
 5957  020014' 001401                             BEQ     LPKTGD            ;ALLOW THIS EXCEPTION
 5958  020016' 000755                             BR      JLPKER            ;ELSE REPORT ERROR
 5959                                                          ;*****DQ11 SPECIAL CODE ENDS HERE
 5960  020020' 016703  001246         LPKTGD:    MOV     LPKPKP,R3
 5961  020024' 012702  000004                     MOV     #4,R2
 5962  020030' 160102                             SUB     R1,R2
 5963  020032' 112023                 LPKLPG:    MOVB    (R0)+,(R3)+       ;STORE TAG IN
 5964  020034' 005301                             DEC     R1                ;PACKED CODE
 5965  020036' 001375                             BNE     LPKLPG
 5966  020040' 005702                             TST     R2                ;IF 4 CHARS ALREADY MOVED
 5967  020042' 001630                             BEQ     LPKAMP            ;GO ADVANCE MODEL POINTER
 5968  020044' 112723  000040         LPKLPB:    MOVB    #40,(R3)+         ;FILL IN TRAILING BLANKS
 5969  020050' 005302                             DEC     R2
 5970  020052' 001374                             BNE     LPKLPB
 5971  020054' 000623                             BR      LPKAMP            ;THEN ADVANCE MODEL PTR
 5972  020056' 016767  006542  006544  LPKSPH:    MOV     LSCNPT,LSPTSV     ;SAVE SCAN POINTER
 5973  020064' 122712  000247                     CMPB    #247,(R2)
 5974  020070' 001422                             BEQ     LPKSRS            ;GO HANDLE SET/RESET
 5975  020072' 122712  000250                     CMPB    #250,(R2)
 5976  020076' 001417                             BEQ     LPKSRS            ;GO HANDLE SET/RESET
 5977  020100' 122712  000100                     CMPB    #100,(R2)
 5978  020104' 001476                             BEQ     LPKIF             ;GO HANDLE IF
 5979  020106' 122712  000150                     CMPB    #150,(R2)
 5980  020112' 001002                             BNE     LPKSJ1
 5981  020114' 000167  000444                     JMP     LPKPRT            ;GO HANDLE PRINT
 5982  020120' 122712  000170         LPKSJ1:    CMPB    #170,(R2)
 5983  020124' 001002                             BNE     LPKEJ1
 5984  020126' 000167  000720                     JMP     LPKFIL            ;GO HANDLE FILL
 5985  020132' 000167  177272         LPKEJ1:    JMP     LPKERR
 5986  020136' 000167  177014         LPKSRS:    JMP     LPKSVP            ;SET MODEL VECTOR POINTER
 5987  020142' 016767  006456  006460  LPKBIT:    MOV     LSCNPT,LSPTSV     ;SAVE SCAN POINTER
 5988  020150' 005004                             CLR     R4                ;SET WORD COUNTER=0
 5989  020152' 004567  006230         LPKBLP:    JSR     R5,SCAN           ;FIND NEXT WORD
 5990  020156' 122710  000015                     CMPB    #15,(R0)          ;WAS IT A CAR. RET
 5991  020162' 001412                             BEQ     LPKPBT            ;YES-DONE
 5992  020164' 005301                             DEC     R1
 5993  020166' 160100                             SUB     R1,R0
 5994  020170' 122710  000123                     CMPB    #'S,(R0)          ;WAS WORD SET ?
 5995  020174' 001405                             BEQ     LPKPBT            ;YES-DONE
 5996  020176' 122710  000103                     CMPB    #'C,(R0)          ;WAS WORD CLEAR ?
 5997  020202' 001402                             BEQ     LPKPBT            ;YES-DONE
 5998  020204' 005204                             INC     R4                ;NO-INCREMENT WORD COUNT
 5999  020206' 000761                             BR      LPKBLP            ;AND SCAN NEXT WORD
 6000  020210' 016767  006414  006406  LPKPBT:    MOV     LSPTSV,LSCNPT     ;RESTORE SCAN POINTER
 6001  020216' 010400                             MOV     R4,R0
 6002  020220' 004567  005070                     JSR     R5,BTPACK         ;SET BITS IN 16 BIT WORD
```

```
 6003  020224' 177204                          .WORD   LPKERR-.
 6004  020226' 010023                          MOV     R0,(R3)+        ;STORE IT IN PACKED CODE
 6005  020230' 016767  006370  006372  LPKSR1: MOV     LSCNPT,LSPTSV   ;SAVE SCAN POINTER
 6006  020236' 004567  006144                  JSR     R5,SCAN         ;SCAN NEXT WORD
 6007  020242' 122710  000015                  CMPB    #15,(R0)        ;CARRIAGE RET ?
 6008  020246' 001005                          BNE     LPKSR3          ;NO-KEEP SEARCHING
 6009  020250' 016767  006354  006346  LPKSR2: MOV     LSPTSV,LSCNPT   ;RESTORE SCAN POINTER
 6010  020256' 000167  177242                  JMP     LPKAMP          ;GO ADVANCE MODEL POINTER
 6011  020262' 126027  177777  000124  LPKSR3: CMPB    -1(R0),#'T
 6012  020270' 001357                          BNE     LPKSR1
 6013  020272' 121027  000117                  CMPB    (R0),#'O        ;LAST 2 LETTERS 'TO' ?
 6014  020276' 001354                          BNE     LPKSR1          ;NO-KEEP SEARCHING
 6015  020300' 000763                          BR      LPKSR2          ;YES-GO PROCESS BRANCH
 6016  020302' 004567  006100          LPKIF:  JSR     R5,SCAN
 6017  020306' 004567  006074                  JSR     R5,SCAN         ;SCAN AHEAD
 6018  020312' 005301                          DEC     R1
 6019  020314' 160100                          SUB     R1,R0
 6020  020316' 122710  000102                  CMPB    #'B,(R0)        ;LAST WORD 'BIT'
 6021  020322' 001475                          BEQ     LPKBT1          ;YES-ANALYZE IF BIT INSTR.
 6022  020324' 122710  000101                  CMPB    #'A,(R0)        ;LAST WORD 'AT'
 6023  020330' 001010                          BNE     LPKCRL          ;YES CHECK RELATIONSHIP
 6024  020332' 062713  004000                  ADD     #4000,(R3)      ;ADD 8 TO FUNCTION CODE
 6025  020336' 004567  006044                  JSR     R5,SCAN
 6026  020342' 004567  006040                  JSR     R5,SCAN         ;SCAN AHEAD
 6027  020346' 005301                          DEC     R1
 6028  020350' 160100                          SUB     R1,R0
 6029  020352' 020127  000001          LPKCRL: CMP     R1,#1           ;CHECK WORD COUNT
 6030  020356' 001422                          BEQ     LPKWD2          ;BRANCH IF TWO
 6031  020360' 103401                          BLO     LPKWD1          ;BRANCH IF ONE
 6032  020362' 000440                          BR      LPKEJ2          ;ERROR IF 3 OR MORE
 6033  020364' 062713  001000          LPKWD1: ADD     #1000,(R3)      ;ADD 2 TO FUNCTION CODE
 6034  020370' 122710  000075                  CMPB    #'=,(R0)        ;WAS IT EQUAL
 6035  020374' 001435                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6036  020376' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6037  020402' 122710  000076                  CMPB    #'>,(R0)        ;WAS IT GTR THAN ?
 6038  020406' 001430                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6039  020410' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6040  020414' 122710  000074                  CMPB    #'<,(R0)        ;WAS IT LESS THAN ?
 6041  020420' 001423                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6042  020422' 000420                          BR      LPKEJ2          ;ERROR IF NONE OF ABOVE
 6043  020424' 062713  002400          LPKWD2: ADD     #2400,(R3)      ;ADD 5 TO FUNCTION CODE
 6044  020430' 152010                          BISB    (R0)+,(R0)      ;COMBINE=AND > ETC.
 6045  020432' 122710  000077                  CMPB    #077,(R0)       ;WAS IT => OR >=
 6046  020436' 001414                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6047  020440' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6048  020444' 122710  000075                  CMPB    #075,(R0)       ;WAS IT <= OR =<
 6049  020450' 001407                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6050  020452' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6051  020456' 122710  000076                  CMPB    #076,(R0)       ;WAS IT <> OR ><
 6052  020462' 001402                          BEQ     LPKRFD          ;YES-RELATION FOUND
 6053  020464' 000167  176740          LPKEJ2: JMP     LPKERR          ;ERROR IF NONE OF ABOVE
 6054  020470' 126327  000001  000112  LPKRFD: CMPB    1(R3),#112      ;IF FC 112 OR LESS
 6055  020476' 101405                          BLOS    LPKRFJ          ;CONTINUE
 6056  020500' 126327  000001  000117          CMPB    1(R3),#117      ;IF FC 117 OR MORE
 6057  020506' 103001                          BHIS    LPKRFJ          ;CONTINUE
 6058  020510' 000765                          BR      LPKEJ2          ;OTHERWISE REPORT ERROR
```

```
 6059  020512' 000167  176440          LPKRFJ: JMP     LPKSVP          ;GO SET MODEL VECTOR PTR
 6060  020516' 004567  005664          LPKBT1: JSR     R5,SCAN         ;SCAN IF BIT STRING
 6061  020522' 122710  000015                  CMPB    #15,(R0)        ;CHECK FOR CAR RET
 6062  020526' 001001                          BNE     LPKTG4
 6063  020530' 000755                          BR      LPKEJ2          ;ERROR IF FOUND
 6064  020532' 005301                  LPKTG4: DEC     R1
 6065  020534' 160100                          SUB     R1,R0           ;POINTER TO FIRST CHAR.
 6066  020536' 122710  000103                  CMPB    #'C,(R0)
 6067  020542' 001404                          BEQ     LPKTG5          ;BRANCH IF C (CLEAR)
 6068  020544' 122710  000123                  CMPB    #'S,(R0)
 6069  020550' 001403                          BEQ     LPKTG6          ;BRANCH IF S (SET)
 6070  020552' 000761                          BR      LPKBT1          ;SCAN NEXT WORD
 6071  020554' 105263  000001          LPKTG5: INCB    1(R3)           ;INCREMENT FUNCTION CODE
 6072  020560' 000167  176372          LPKTG6: JMP     LPKSVP          ;GO SET MODEL VECTOR PTR
 6073
 6074  020564' 004567  005616          LPKPRT: JSR     R5,SCAN
 6075  020570' 122710  000052                  CMPB    #'*,(R0)        ;IS FIRST CHAR A *
 6076  020574' 001502                          BEQ     LPKTXT          ;YES-GO PACK TEXT
 6077  020576' 122710  000043                  CMPB    #'#,(R0)        ;IS FIRST CHAR A #
 6078  020602' 001474                          BEQ     LPKTNR          ;YES - PACK TEXT WITH NO RETN
 6079  020604' 004567  005576                  JSR     R5,SCAN         ;SCAN SECOND WORD
 6080  020610' 022701  000002                  CMP     #2,R1           ;DOES LENGTH=2
 6081  020614' 001013                          BNE     LPKTG7          ;NO-SCAN MORE
 6082  020616' 005301                          DEC     R1
 6083  020620' 160100                          SUB     R1,R0
 6084  020622' 122027  000111                  CMPB    (R0)+,#'I       ;1ST CHAR AN I?
 6085  020626' 001006                          BNE     LPKTG7          ;NO-SCAN MORE
 6086  020630' 121027  000116                  CMPB    (R0),#'N        ;2ND CHAR AN N?
 6087  020634' 001003                          BNE     LPKTG7          ;NO-SCAN MORE
 6088  020636' 062713  002000                  ADD     #2000,(R3)      ;INCR F.C. BY 4
 6089  020642' 000422                          BR      LPKTGA
 6090  020644' 004567  005536          LPKTG7: JSR     R5,SCAN         ;SCAN NEXT WORD
 6091  020650' 122710  000015                  CMPB    #15,(R0)        ;CARRIAGE RETURN?
 6092  020654' 001001                          BNE     LPKTG8          ;NO-CONTINUE
 6093  020656' 000702                          BR      LPKEJ2          ;YES-ERROR
 6094  020660' 004567  005522          LPKTG8: JSR     R5,SCAN         ;SCAN NEXT WORD
 6095  020664' 005301                          DEC     R1
 6096  020666' 160100                          SUB     R1,R0
 6097  020670' 122027  000111                  CMPB    (R0)+,#'I       ;1ST CHAR AN I?
 6098  020674' 001003                          BNE     LPKTG9          ;NO-ERROR
 6099  020676' 121027  000116                  CMPB    (R0),#'N        ;2ND CHAR AN N?
 6100  020702' 001402                          BEQ     LPKTGA          ;YES-CONTINUE
 6101  020704' 000167  176520          LPKTG9: JMP     LPKERR          ;NO-ERROR
 6102  020710' 004567  005472          LPKTGA: JSR     R5,SCAN         ;SCAN NEXT WORD
 6103  020714' 005301                          DEC     R1
 6104  020716' 160100                          SUB     R1,R0
 6105  020720' 121027  000107                  CMPB    (R0),#'G
 6106  020724' 001661                          BEQ     LPKRFD          ;BRANCH IF G (GRAPHICS)
 6107  020726' 121027  000101                  CMPB    (R0),#'A
 6108  020732' 001656                          BEQ     LPKRFD          ;BRANCH IF A (ASCII)
 6109  020734' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6110  020740' 121027  000117                  CMPB    (R0),#'O
 6111  020744' 001651                          BEQ     LPKRFD          ;BRANCH IF O (OCTAL)
 6112  020746' 105263  000001                  INCB    1(R3)           ;INCREMENT F.C.
 6113  020752' 121027  000104                  CMPB    (R0),#'D
 6114  020756' 001644                          BEQ     LPKRFD          ;BRANCH IF D (DECIMAL)
```

```
6115  020760' 105263  000001                  INCB    1(R3)                   ;INCREMENT F.C.
6116  020764' 121027  000102                  CMPB    (R0),#'B
6117  020770' 001637                          BEQ     LPKRFD                  ;BRANCH IF B (BINARY)
6118  020772' 000744                          BR      LPKTG9                  ;ERROR IF NONE OF ABOVE
6119  020774' 062713  044400          LPKTNR: ADD     #44400,(R3)             ;ADD 111 TO F.C.
6120  021000' 000402                          BR      LPKADP                  ;GO ADJUST PACK POINTER
6121  021002' 062713  034400          LPKTXT: ADD     #34400,(R3)             ;ADD 71 TO F.C.
6122  021006' 062703  000006          LPKADP: ADD     #6,R3                   ;ADJUST PACK POINTER
6123  021012' 005200                  LPKITX: INC     R0
6124  021014' 112023                  LPKLP8: MOVB    (R0)+,(R3)+             ;MOVE WORD TO PACKED CODE
6125  021016' 122760  000015  177777          CMPB    #15,-1(R0)              ;FOUND CARRIAGE RET ?
6126  021024' 001373                          BNE     LPKLP8                  ;NO-MOVE ANOTHER BYTE
6127  021026' 032703  000001          LPKEVN: BIT     #1,R3                   ;NEXT ADDRESS EVEN ?
6128  021032' 001405                          BEQ     LPKTXD                  ;YES - TEXT DONE
6129  021034' 112763  000040  177777          MOVB    #40,-1(R3)              ;STORE BLANK IN PLACE OF CR
6130  021042' 112723  000015                  MOVB    #15,(R3)+               ;STORE CR TO MAKE IT EVEN
6131  021046' 000167  176362          LPKTXD: JMP     LPKEXT                  ;YES-EXIT
6132  021052' 005004                  LPKFIL: CLR     R4                      ;SET WORD COUNT=0
6133  021054' 004567  005326          LPKFL1: JSR     R5,SCAN                 ;SCAN A WORD
6134  021060' 005204                          INC     R4                      ;INCREMENT WORD COUNT
6135  021062' 005301                  LPKFT1: DEC     R1
6136  021064' 160100                          SUB     R1,R0
6137  021066' 121027  000122                  CMPB    (R0),#'R                ;1ST CHAR R (RANDOM)
6138  021072' 001415                          BEQ     LPKFL2                  ;YES-BRANCH
6139  021074' 121027  000107                  CMPB    (R0),#'G                ;1ST CHAR G (GRAPHICS)
6140  021100' 001424                          BEQ     LPKFL3                  ;YES-BRANCH
6141  021102' 121027  000101                  CMPB    (R0),#'A                ;1ST CHAR A (ASCII) ?
6142  021106' 001421                          BEQ     LPKFL3                  ;YES - BRANCH
6143  021110' 121027  000120                  CMPB    (R0),#'P                ;1ST CHAR P (PATTERN)
6144  021114' 001434                          BEQ     LPKFL5
6145  021116' 121027  000015                  CMPB    (R0),#15                ;CARRIAGE RETURN?
6146  021122' 001425                          BEQ     LPKFL4                  ;YES-BRANCH
6147  021124' 000753                          BR      LPKFL1                  ;SCAN SOME MORE
6148  021126' 004567  005254          LPKFL2: JSR     R5,SCAN
6149  021132' 005204                          INC     R4                      ;INCREMENT WORD COUNT
6150  021134' 121027  000015                  CMPB    (R0),#15                ;LAST WORD ?
6151  021140' 001350                          BNE     LPKFT1                  ;NO-DISREGARD
6152  021142' 022704  000004                  CMP     #4,R4                   ;WORD COUNT=4
6153  021146' 001446                          BEQ     LPKFA1                  ;YES
6154  021150' 000443                          BR      LPKFA2                  ;NO
6155  021152' 004567  005230          LPKFL3: JSR     R5,SCAN
6156  021156' 005204                          INC     R4                      ;INCREMENT WORD COUNT
6157  021160' 121027  000015                  CMPB    (R0),#15                ;LAST WORD ?
6158  021164' 001336                          BNE     LPKFT1                  ;NO-DISREGARD
6159  021166' 022704  000004                  CMP     #4,R4                   ;WORD COUNT=4
6160  021172' 001430                          BEQ     LPKFA3                  ;YES
6161  021174' 000425                          BR      LPKFA4                  ;NO
6162  021176' 022704  000004          LPKFL4: CMP     #4,R4                   ;WORD COUNT=4
6163  021202' 001420                          BEQ     LPKFA5                  ;YES
6164  021204' 000415                          BR      LPKFA6
6165  021206' 004567  005174          LPKFL5: JSR R5,SCAN
6166  021212' 005204                          INC     R4                      ;INCREMENT WORD COUNT
6167  021214' 121027  000015                  CMPB    (R0),#15                ;LAST WORD ?
6168  021220' 001320                          BNE     LPKFT1                  ;NO - DISREGARD
6169  021222' 022704  000004                  CMP     #4,R4                   ;WORD COUNT = 4?
6170  021226' 001402                          BEQ     LPKFA7                  ;YES
```

```
6171  021230' 105263 000001                 INCB    1(R3)                   ;NO
6172  021234' 105263 000001          LPKFA7: INCB    1(R3)
6173  021240' 105263 000001          LPKFA6: INCB    1(R3)                   ;ADD
6174  021244' 105263 000001          LPKFA5: INCB    1(R3)                   ;APPROPRIATE
6175  021250' 105263 000001          LPKFA4: INCB    1(R3)                   ;AMOUNT
6176  021254' 105263 000001          LPKFA3: INCB    1(R3)                   ;TO
6177  021260' 105263 000001          LPKFA2: INCB    1(R3)                   ;FUNCTION
6178  021264'                        LPKFA1:                                 ;CODE
6179  021264' 000167 175666                 JMP     LPKSVP                  ;GO SET VECTOR POINTER
6180
6181
6182  021270' 000000                 LPKPTP: .WORD   0                       ;P TBL POINTER SAVE
6183  021272' 000000                 LPKPKP: .WORD   0                       ;PACK POINTER SAVE
6184  021274' 000000                 LBDLIN: .WORD   0                       ;BAD LINE NBR FLAG
6185  021276' 000000                 LPKIFL: .WORD   0                       ;INTERNAL TBL FLG
6186  021300' 000000                 LMSTBE: .WORD   0                       ;MODEL STMNT TBL END
6187
6188
6189
6190  021302' 047522 040524 042524  LPKTBL: .ASCII  /ROTATE/                 ;PACK/UNPACK TABLE
6191  021310'    000    000                 .BYTE   0,0
6192  021312' 020040 047111 051103          .ASCII  /  INCR/
6193  021320'    010    000                 .BYTE   010,0
6194  021322' 020040 042504 051103          .ASCII  /  DECR/
6195  021330'    020    000                 .BYTE   020,0
6196  021332' 020040 047515 042526          .ASCII  /  MOVE/
6197  021340'    030    000                 .BYTE   030,0
6198  021342' 020040 040440 042104          .ASCII  /   ADD/
6199  021350'    040    000                 .BYTE   040,0
6200  021352' 020040 051440 041125          .ASCII  /   SUB/
6201  021360'    050    000                 .BYTE   050,0
6202  021362' 020040 020040 043111          .ASCII  /    IF/
6203  021370'    100    377                 .BYTE   100,377
6204  021372' 020040 020040 043111          .ASCII  /    IF/
6205  021400'    110    377                 .BYTE   110,377
6206  021402' 020040 042522 042101          .ASCII  /  READ/
6207  021410'    120    000                 .BYTE   120,0
6208  021412' 053440 044522 042524          .ASCII  / WRITE/
6209  021420'    130    000                 .BYTE   130,0
6210  021422' 042526 044522 054506          .ASCII  /VERIFY/
6211  021430'    140    000                 .BYTE   140,0
6212  021432' 050040 044522 052116          .ASCII  / PRINT/
6213  021440'    150    377                 .BYTE   150,377
6214  021442' 041040 042522 045501          .ASCII  / BREAK/
6215  021450'    160    000                 .BYTE   160,0
6216  021452' 020040 044506 046114          .ASCII  /  FILL/
6217  021460'    170    377                 .BYTE   170,377
6218  021462' 020040 042440 042116          .ASCII  /   END/
6219  021470'    240    000                 .BYTE   240,0
6220  021472' 050040 044522 052116          .ASCII  / PRINT/
6221  021500'    241    000                 .BYTE   241,0
6222  021502' 020040 047514 042101          .ASCII  /  LOAD/
6223  021510'    242    000                 .BYTE   242,0
6224  021512' 042040 046105 054501          .ASCII  / DELAY/
6225  021520'    243    000                 .BYTE   243,0
6226  021522' 043440 020117 047524          .ASCII  / GO TO/
```

```
6227  021530'    244    000                    .BYTE   244,0
6228  021532' 020040 047507 047524             .ASCII  /  GOTO/
6229  021540'    244    000                    .BYTE   244,0
6230  021542' 020040 044514 045516             .ASCII  /  LINK/
6231  021550'    245    000                    .BYTE   245,0
6232  021552' 042522 052524 047122             .ASCII  /RETURN/
6233  021560'    246    000                    .BYTE   246,0
6234  021562' 020040 051440 052105             .ASCII  /   SET/
6235  021570'    247    377                    .BYTE   247,377
6236  021572' 041440 042514 051101             .ASCII  / CLEAR/
6237  021600'    250    377                    .BYTE   250,377
6238           001                    .IF NDF MM
6239  021602' 042526 052103 051117             .ASCII  /VECTOR/
6240  021610'    251    000                    .BYTE   251,0
6241           000                    .ENDC
6242  021612' 046040 052105 047507             .ASCII  / LETGO/
6243  021620'    252    000                    .BYTE   252,0
6244  021622' 042440 052116 054522             .ASCII  / ENTRY/
6245  021630'    253    000                    .BYTE   253,0
6246  021632' 020040 054105 052111             .ASCII  /  EXIT/
6247  021640'    254    000                    .BYTE   254,0
6248           001                    .IF NDF MM
6249  021642' 051040 051505 052105             .ASCII  / RESET/
6250  021650'    255    000                    .BYTE   255,0
6251           000                    .ENDC
6252  021652' 020040 047504 042516             .ASCII  /  DONE/
6253  021660'    256    000                    .BYTE   256,0
6254  021662' 042504 042514 042524             .ASCII  /DELETE/
6255  021670'    257    000                    .BYTE   257,0
6256  021672' 042516 040507 042524             .ASCII  /NEGATE/
6257  021700'    260    000                    .BYTE   260,0
6258  021702' 050040 044522 052116             .ASCII  / PRINT/
6259  021710'    261    000                    .BYTE   261,0
6260  021712' 050040 052501 042523             .ASCII  / PAUSE/
6261  021720'    262    000                    .BYTE   262,0
6262  021722' 020040 047527 042122             .ASCII  /  WORD/
6263  021730'    263    000                    .BYTE   263,0
6264  021732' 020040 020040 020040             .ASCII  /     /
6265  021740'    377    000                    .BYTE   377,0                   ;END OF TABLE
6266  021742' 000006                           .BLKB   6
6267  021750' 000000                 LPKTBE:  .WORD   0
6268
6269
6270
6271                                    ;
6272                                    ;       MODEL STATEMENT VECTOR TABLE
6273                                    ;
6274  021752' 000242 003046             LMPNTR:  .WORD   242,LLOAD-LBASE1
6275  021756' 000011 003055                      .WORD   011,LINCR1-LBASE1
6276  021762' 000021 003055                      .WORD   021,LDECR1-LBASE1
6277  021766' 000013 003057                      .WORD   013,LINCR2-LBASE1
6278  021772' 000023 003057                      .WORD   023,LDECR2-LBASE1
6279  021776' 000030 003632                      .WORD   030,LMOVE0-LBASE1
6280  022002' 000033 003064                      .WORD   033,LMOVE1-LBASE1
6281  022006' 000035 003071                      .WORD   035,LMOVE2-LBASE1
6282  022012' 000043 003064                      .WORD   043,LADD1-LBASE1
```

```
6283  022016' 000053  003101                .WORD   053,LSUB1-LBASE1
6284  022022' 000045  003071                .WORD   045,LADD2-LBASE1
6285  022026' 000055  003110                .WORD   055,LSUB2-LBASE1
6286  022032' 000260  003055                .WORD   260,LNEGAT-LBASE1
6287  022036' 000247  003122                .WORD   247,LSET-LBASE1
6288  022042' 000250  003122                .WORD   250,LCLEAR-LBASE1
6289  022046' 000100  003130                .WORD   100,LIFST-LBASE1
6290  022052' 000101  003150                .WORD   101,LIFCLR-LBASE1
6291  022056' 000102  003172                .WORD   102,LIFEQ1-LBASE1
6292  022062' 000103  003204                .WORD   103,LIFGT1-LBASE1
6293  022066' 000104  003216                .WORD   104,LIFLT1-LBASE1
6294  022072' 000105  003230                .WORD   105,LIFGE1-LBASE1
6295  022076' 000106  003243                .WORD   106,LIFLE1-LBASE1
6296  022102' 000107  003256                .WORD   107,LIFNE1-LBASE1
6297  022106' 000112  003271                .WORD   112,LIFEQ2-LBASE1
6298  022112' 000117  003306                .WORD   117,LIFNE2-LBASE1
6299  022116' 000244  003324                .WORD   244,LGOTO-LBASE1
6300  022122' 000245  003324                .WORD   245,LLINK-LBASE1
6301  022126' 000246  003326                .WORD   246,LRETN-LBASE1
6302  022132' 000241  003330                .WORD   241,LPRNI-LBASE1
6303  022136' 000261  003333                .WORD   261,LPRNNR-LBASE1
6304  022142' 000154  003336                .WORD   154,LPRNG1-LBASE1
6305  022146' 000155  003351                .WORD   155,LPRNO1-LBASE1
6306  022152' 000156  003364                .WORD   156,LPRND1-LBASE1
6307  022156' 000157  003401                .WORD   157,LPRNB1-LBASE1
6308  022162' 000150  003415                .WORD   150,LPRNG2-LBASE1
6309  022166' 000151  003433                .WORD   151,LPRNO2-LBASE1
6310  022172' 000152  003451                .WORD   152,LPRND2-LBASE1
6311  022176' 000153  003471                .WORD   153,LPRNB2-LBASE1
6312  022202' 000161  003055                .WORD   161,LBRK1-LBASE1
6313  022206' 000163  003704                .WORD   163,LBRK2-LBASE1
6314  022212' 000170  003510                .WORD   170,LFILR1-LBASE1
6315  022216' 000172  003526                .WORD   172,LFILG1-LBASE1
6316  022222' 000174  003543                .WORD   174,LFILV1-LBASE1
6317  022226' 000171  003552                .WORD   171,LFILR2-LBASE1
6318  022232' 000173  003573                .WORD   173,LFILG2-LBASE1
6319  022236' 000175  003613                .WORD   175,LFILV2-LRASE1
6320  022242' 000176  003543                .WORD   176,LFILP1-LBASE1
6321  022246' 000177  003613                .WORD   177,LFILP2-LBASE1
6322  022252' 000001  003055                .WORD   001,LROT1-LBASE1
6323  022256' 000003  003625                .WORD   003,LROT2-LBASE1
6324  022262' 000243  003055                .WORD   243,LDELAY-LBASE1
6325  022266' 000120  003632                .WORD   120,LREAD1-LBASE1
6326  022272' 000121  003055                .WORD   121,LREAD2-LBASE1
6327  022276' 000123  003633                .WORD   123,LREAD3-LBASE1
6328  022302' 000125  003642                .WORD   125,LREAD4-LBASE1
6329  022306' 000130  003632                .WORD   130,LWRIT1-LBASE1
6330  022312' 000131  003055                .WORD   131,LWRIT2-LBASE1
6331  022316' 000133  003656                .WORD   133,LWRIT3-LBASE1
6332  022322' 000135  003665                .WORD   135,LWRIT4-LBASE1
6333  022326' 000143  003543                .WORD   143,LVERF1-LBASE1
6334  022332' 000145  003613                .WORD   145,LVERF2-LBASE1
6335           001          .IF NDF MM
6336  022336' 000251  003677                .WORD   251,LVECT-LBASE1
6337           000          .ENDC
6338  022342' 000252  003632                .WORD   252,LETGO-LBASE1
```

```
6339  022346' 000253  003632                    .WORD   253,LENTRY-LBASE1
6340  022352' 000254  003632                    .WORD   254,LEXIT-LBASE1
6341            001                .IF NDF MM
6342  022356' 000255  003632                    .WORD   255,LRESET-LBASE1
6343            000                .ENDC
6344  022362' 000262  003632                    .WORD   262,LPAUSE-LBASE1
6345  022366' 000263  003055                    .WORD   263,LWORD-LBASE1
6346  022372' 000240  003632                    .WORD   240,LEND-LBASE1
6347  022376' 000377  003711                    .WORD   377,LTBEND-LBASE1          ;END OF TABLE
6348
6349
6350
6351                                  ;
6352                                  ;       MODEL STATEMENT TABLE
6353                                  ;
6354  022402' 053777  052111  177510  LLOAD:  .ASCIZ  <377>/WITH/<377>
      022410'    000
6355
6356  022411'                         LWORD:
6357  022411'                         LBRK1:
6358  022411'                         LWRIT2:
6359  022411'                         LINCR1:
6360  022411'                         LNEGAT:
6361  022411'                         LROT1:
6362  022411'                         LDELAY:
6363  022411'                         LREAD2:
6364  022411'     377     000         LDECR1: .BYTE   377,0
6365  022413'                         LINCR2:
6366  022413'     377  054502  000377 LDECR2: .ASCIZ  <377>/BY/<377>
6367  022420'                         LADD1:
6368  022420' 052377  177517     000 LMOVE1: .ASCIZ  <377>/TO/<377>
6369  022425'                         LADD2:
6370  022425'     377  052101  052377 LMOVE2: .ASCIZ  <377>/AT/<377>/TO/<377>
      022432' 177517     000
6371  022435'     377  051106  046517 LSUB1:  .ASCIZ  <377>/FROM/<377>
      022442' 000377
6372  022444' 040777  177524  051106 LSUB2:  .ASCIZ  <377>/AT/<377>/FROM/<377>
      022452' 046517  000377
6373  022456'                         LSET:
6374  022456' 041377  052111  000375 LCLEAR: .ASCIZ  <377>/BIT/<375>
6375  022464' 041377  052111  051775 LIFST:  .ASCIZ  <377>/BIT/<375>/SET GO TO/<376>
      022472' 052105  043440  020117
      022500' 047524  000376
6376  022504' 041377  052111  041775 LIFCLR: .ASCIZ  <377>/BIT/<375>/CLEAR GO TO/<376>
      022512' 042514  051101  043440
      022520' 020117  047524  000376
6377  022526' 036777  043777  020117 LIFEQ1: .ASCIZ  <377>/=/<377>/GO TO/<376>
      022534' 047524  000376
6378  022540' 037377  043777  020117 LIFGT1: .ASCIZ  <377>/>/<377>/GO TO/<376>
      022546' 047524  000376
6379  022552' 036377  043777  020117 LIFLT1: .ASCIZ  <377></<377>/GO TO/<376>
      022560' 047524  000376
6380  022564' 036777  177476  047507 LIFGE1: .ASCIZ  <377>/=>/<377>/GO TO/<376>
      022572' 052040  177117     000
6381  022577'     377  036474  043777 LIFLE1: .ASCIZ  <377></=/<377>/GO TO/<376>
      022604' 020117  047524  000376
```

```
6382  022612' 036377 177476 047507  LIFNE1: .ASCIZ  <377>/(/<377>/GO TO/<376>
      022620' 052040 177117    000
6383  022625'    377 052101 036777  LIFEQ2: .ASCIZ  <377>/AT/<377>/=/<377>/GO TO/<376>
      022632' 043777 020117 047524
      022640' 000376
6384  022642' 040777 177524 037074  LIFNE2: .ASCIZ  <377>/AT/<377>/(/<377>/GO TO/<376>
      022650' 043777 020117 047524
      022656' 000376
6385  022660'                       LGOTO:
6386  022660'    376    000        LLINK:  .BYTE   376,0
6387  022662'    373    000        LRETN:  .BYTE   373,0
6388  022664' 176052    000        LPRNI:  .ASCIZ  /*/<374>
6389  022667'    043 000374        LPRNNR: .ASCIZ  /#/<374>
6390  022672' 044777 020116 051501  LPRNG1: .ASCIZ  <377>/IN ASCII/<373>
      022700' 044503 175511    000
6391  022705'    377 047111 047440  LPRNO1: .ASCIZ  <377>/IN OCTAL/<373>
      022712' 052103 046101 000373
6392  022720' 044777 020116 042504  LPRND1: .ASCIZ  <377>/IN DECIMAL/<373>
      022726' 044503 040515 175514
      022734'    000
6393  022735'    377 047111 041040  LPRNB1: .ASCIZ  <377>/IN BINARY/<373>
      022742' 047111 051101 175531
      022750'    000
6394  022751'    377 052101 044777  LPRNG2: .ASCIZ  <377>/AT/<377>/IN ASCII/<373>
      022756' 020116 051501 044503
      022764' 175511    000
6395  022767'    377 052101 044777  LPRNO2: .ASCIZ  <377>/AT/<377>/IN OCTAL/<373>
      022774' 020116 041517 040524
      023002' 175514    000
6396  023005'    377 052101 044777  LPRND2: .ASCIZ  <377>/AT/<377>/IN DECIMAL/<373>
      023012' 020116 042504 044503
      023020' 040515 175514    000
6397  023025'    377 052101 044777  LPRNB2: .ASCIZ  <377>/AT/<377>/IN BINARY/<373>
      023032' 020116 044502 040516
      023040' 054522 000373
6398  023044' 053777 052111 020110  LFILR1: .ASCIZ  <377>/WITH RANDOM/<373>
      023052' 040522 042116 046517
      023060' 000373
6399  023062' 053777 052111 020110  LFILG1: .ASCIZ  <377>/WITH ASCII/<373>
      023070' 051501 044503 175511
      023076'    000
6400  023077'                       LFILP1:
6401  023077'                       LVERF1:
6402  023077'    377 044527 044124  LFILV1: .ASCIZ  <377>/WITH/<377>
      023104' 000377
6403  023106' 040777 177524 044527  LFILR2: .ASCIZ  <377>/AT/<377>/WITH RANDOM/<373>
      023114' 044124 051040 047101
      023122' 047504 175515    000
6404  023127'    377 052101 053777  LFILG2: .ASCIZ  <377>/AT/<377>/WITH ASCII/<373>
      023134' 052111 020110 051501
      023142' 044503 175511    000
6405  023147'                       LFILP2:
6406  023147'                       LVERF2:
6407  023147'    377 052101 053777  LFILV2: .ASCIZ  <377>/AT/<377>/WITH/<377>
      023154' 052111 177510    000
6408  023161'    377 052101 000377  LROT2:  .ASCIZ  <377>/AT/<377>
```

```
6409  023166'                          LMOVEO:
6410  023166'                          LWRIT1:
6411  023166'                          LREAD1:
6412  023166'                          LETGO:
6413  023166'                          LENTRY:
6414  023166'                          LEXIT:
6415                001                .IF NDF MM
6416  023166'                          LRESET:
6417                000                .ENDC
6418  023166'                          LPAUSE:
6419  023166'     000                  LEND:   .BYTE   0
6420  023167'     377   047111  047524 LREAD3: .ASCIZ  <377>/INTO/<377>
      023174'  000377
6421  023176'  044777  052116  177517 LREAD4: .ASCIZ  <377>/INTO/<377>/FROM/<375>
      023204'  051106  046517  000375
6422  023212'  043377  047522  177515 LWRIT3: .ASCIZ  <377>/FROM/<377>
      023220'     000
6423  023221'     377  051106  046517 LWRIT4: .ASCIZ  <377>/FROM/<377>/TO/<375>
      023226'  052377  176517     000
6424                001                .IF NDF MM
6425  023233'     377  047524  000376 LVECT:  .ASCIZ  <377>/TO/<376>
6426                000                .ENDC
6427  023240'  047777  176516     000 LBRK2:  .ASCIZ  <377>/ON/<375>
6428  023245'     000                  LTBEND: .BYTE   000                     ;END OF TABLE
6429                                             .EVEN
6430
6431
6432  023246'     000                  LDONE:  .BYTE   0                       ;DONE FLAG
6433  023247'     000                  LDELET: .BYTE   0                       ;DELETE FLAG (MUST FOLLOW LDONE)
6434  023250'     000                  LIGNOR: .BYTE   0                       ;IGNORE FLAG
6435  023251'     000                  LERROR: .BYTE   0                       ;ERROR FLAG
6436
6437  023252'  047504  042516         LASDON: .ASCII  /DONE/
6438  023256'  000000                  LPKEPT: .WORD   0
6439  023260'  000000                  LPKPSV: .WORD   0
6440  023262'  000000                  LPKMPS: .WORD   0
6441
6442
6443  023264'  000110                  LPKBUF: .BLKB   72.                     ;PACK BUFFER
```

```
                                .SBTTL  UNPACK SUBROUTINE

                        ;*******************************************************************
                        ;
                        ;       THIS SUBROUTINE EXPANDS ONE LINE OF
                        ;       PACKED CODE BACK TO A FORMAT SIMILAR
                        ;       TO THE USERS ORIGINAL ENTRY.
                        ;
                        ;       CALLING SEQUENCE - JSR  R5,UNPACK
                        ;
                        ;       ENTRY -  LUPKPT  POINTS TO PACKED CODE.
                        ;
                        ;       EXIT -  R0 CONTAINS UNPACKED CHARACTER COUNT
                        ;               R1 AND LUPKPT POINT TO NEXT PACKED ENTRY
                        ;
                        ;*******************************************************************

023374' 010346          UNPACK: MOV     R3,-(SP)
023376' 010446                  MOV     R4,-(SP)                ;SAVE R3, R4 AND R5
023400' 010546                  MOV     R5,-(SP)
023402' 005067  000702          CLR     LENDFL                  ;RESET END FLAG
023406' 010705                  MOV     PC,R5
023410' 062705  010763          ADD     #LINBUF+73.-.,R5        ;SET OUTPUT POINTER
023414' 012700  000111          MOV     #73.,R0
023420' 112745  000040  LUPLP1: MOVB    #40,-(R5)               ;CLEAR OUTPUT LINE BFR
023424' 005300                  DEC     R0
023426' 001374                  BNE     LUPLP1
023430' 016704  000660          MOV     LUPKPT,R4               ;POINT R4 AT PACKED CODE
023434' 112467  000651          MOVB    (R4)+,LPLNTH            ;GET LENGTH
023440' 111403                  MOVB    (R4),R3                 ;GET FUNCTION CODE
023442' 112714  000377          MOVB    #377,(R4)               ;PUT NULL (ONES) FOR F.C.
023446' 010400                  MOV     R4,R0
023450' 010501                  MOV     R5,R1
023452' 004567  015346          JSR     R5,DECASC               ;CONVERT LINE # TO ASCII
023456' 110314                  MOVB    R3,(R4)                 ;RESTORE F.C.
023460' 010105                  MOV     R1,R5
023462' 010004                  MOV     R0,R4
023464' 012467  000622          MOV     (R4)+,LPHYSL            ;GET AND SAVE PHYSICAL ADDRESS
023470' 112725  000011          MOVB    #11,(R5)+               ;STORE A TAB
023474' 010700                  MOV     PC,R0
023476' 062700  176254          ADD     #LMPNTR-.,R0            ;SET R0 TO MODEL VECTOR TBL
023502' 010302                  MOV     R3,R2                   ;SAVE FUNCTION CODE
023504' 122710  000377  LUPLP2: CMPB    #377,(R0)
023510' 001405                  BEQ     LUPSEV                  ;GO SET EXTERNAL VECTOR
023512' 120310                  CMPB    R3,(R0)
023514' 001437                  BEQ     LUPSMP
023516' 062700  000004          ADD     #4,R0
023522' 000770                  BR      LUPLP2
023524' 016700  000566  LUPSEV: MOV     LUPPTP,R0               ;DETERMINE END
023530' 062700  000354          ADD     #PTLGTH+DVCTEP,R0
023534' 061000                  ADD     (R0),R0                 ;OF EXT VEC TBL
023536' 010067  175536          MOV     R0,LMSTBE
023542' 016700  000550          MOV     LUPPTP,R0               ;POINT R0 AT START
023546' 062700  000352          ADD     #PTLGTH+DVVTEP,R0       ;OF
023552' 061000                  ADD     (R0),R0                 ;EXTERN VECTOR TBL
```

```
 6501  023554' 020067  175520          10$:    CMP     R0,LMSTBE               ;R0 AT TBL END?
 6502  023560' 001520                          BEQ     LUPEXT                  ;YES - EXIT
 6503  023562' 121003                          CMPB    (R0),R3                 ;CHECK F.C. IN TBL
 6504  023564' 001403                          BEQ     LUPSEP                  ;BRANCH IF FOUND
 6505  023566' 062700  000004                  ADD     #4,R0                   ;BUMP POINTER
 6506  023572' 000770                          BR      10$                     ;CHECK NEXT ENTRY
 6507  023574' 062700  000002  LUPSEP: ADD     #2,R0
 6508  023600' 016703  000512                  MOV     LUPPTP,R3
 6509  023604' 062703  000242                  ADD     #PTLGTH,R3
 6510  023610' 061003                          ADD     (R0),R3                 ;SET EXTERN MDL POINTER
 6511  023612' 000406                          BR      LUPTMF                  ;GO TEST MULT FORMAT
 6512  023614' 062700  000002  LUPSMP: ADD     #2,R0                   ;R0 CONTAINS MODEL ADDR
 6513  023620' 011003                          MOV     (R0),R3
 6514  023622' 060703                          ADD     PC,R3
 6515  023624' 062703  173510  LBASE2: ADD     #LBASE1-LBASE2,R3       ;R3 NOW POINTS TO MODEL
 6516  023630' 120227  000237  LUPTMF: CMPB    R2,#237
 6517  023634' 101002                          BHI     LUPNPI
 6518  023636' 042702  000007                  BIC     #7,R2
 6519  023642' 010701          LUPNPI: MOV     PC,R1
 6520  023644' 062701  175444                  ADD     #LPKTBL+6-.,R1          ;POINT R1 AT PACK TBL
 6521  023650' 122711  000377  LUPLP3: CMPB    #377,(R1)
 6522  023654' 001405                          BEQ     LUPSEX                  ;EXIT IF F.C. NOT FOUND
 6523  023656' 120211                          CMPB    R2,(R1)
 6524  023660' 001433                          BEQ     LUPIFD                  ;BRANCH IF INSTR. FOUND
 6525  023662' 062701  000010                  ADD     #8.,R1
 6526  023666' 000770                          BR      LUPLP3
 6527  023670' 016701  000422  LUPSEX: MOV     LUPPTP,R1               ;DETERMINE END
 6528  023674' 062701  000352                  ADD     #PTLGTH+DVVTEP,R1       ;OF
 6529  023700' 061101                          ADD     (R1),R1                 ;EXT PACK TABLE
 6530  023702' 062701  000006                  ADD     #6,R1                   ;ADJUST R1
 6531  023706' 010167  176036                  MOV     R1,LPKTBE
 6532  023712' 016701  000400                  MOV     LUPPTP,R1               ;POINT R1
 6533  023716' 062701  000350                  ADD     #PTLGTH+DVPTEP,R1       ;AT START
 6534  023722' 061101                          ADD     (R1),R1                 ;OF EXT PACK TABLE
 6535  023724' 062701  000006                  ADD     #6,R1                   ;ADJUST R1
 6536  023730' 020167  176014          10$:    CMP     R1,LPKTBE               ;R1 AT TBL END?
 6537  023734' 001432                          BEQ     LUPEXT                  ;YES - EXIT
 6538  023736' 120211                          CMPB    R2,(R1)
 6539  023740' 001403                          BEQ     LUPIFD                  ;BRANCH IF INSTR FND
 6540  023742' 062701  000010                  ADD     #8.,R1
 6541  023746' 000770                          BR      10$                     ;ELSE KEEP LOOKING
 6542  023750' 012700  000006  LUPIFD: MOV     #6,R0
 6543  023754' 160001                          SUB     R0,R1                   ;BACK UP TO INST WORD
 6544  023756' 122711  000040  LUPLP4: CMPB    #40,(R1)                ;CHAR A BLANK?
 6545  023762' 001004                          BNE     LUPSIN                  ;NO-STORE INST WORD
 6546  023764' 005201                          INC     R1                      ;BUMP R1
 6547  023766' 005300                          DEC     R0                      ;DECREMENT CHARA COUNT
 6548  023770' 001372                          BNE     LUPLP4                  ;LOOP IF ANY REMAINING
 6549  023772' 000413                          BR      LUPEXT                  ;EXIT IF ALL BLANKS
 6550  023774' 112125          LUPSIN: MOVB    (R1)+,(R5)+             ;STORE INSTRUCTION WORD
 6551  023776' 005300                          DEC     R0
 6552  024000' 001375                          BNE     LUPSIN
 6553  024002' 112725  000040                  MOVB    #40,(R5)+               ;STORE A BLANK
 6554  024006' 122702  000240                  CMPB    #240,R2                 ;WAS IT AN 'END'
 6555  024012' 001035                          BNE     LUPEMB                  ;NO-CONTINUE
 6556  024014' 012767  000001  000266          MOV     #1,LENDFL               ;YES-SET END FLAG
```

```
6557  024022' 112725 000015          LUPEXT: MOVB   #15,(R5)+          ;STORE A CR/LF
6558  024026' 112725 000012                  MOVB   #12,(R5)+
6559  024032' 010700                         MOV    PC,R0
6560  024034' 062700 010226                  ADD    #LINBUF-.,R0
6561  024040' 160005                         SUB    R0,R5
6562  024042' 010500                         MOV    R5,R0              ;R0=UNPACKED CHAR COUNT
6563  024044' 116701 000241                  MOVB   LPLNTH,R1
6564  024050' 001003                         BNE    LPLNNZ             ;CONTINUE IF LENGTH NOT 0
6565  024052' 112767 000001 000230           MOVB   #1,LENDFL          ;OTHERWISE SET END FLAG
6566  024060' 060167 000230          LPLNNZ: ADD    R1,LUPKPT          ;POINT LUPKPT AND
6567  024064' 016701 000224                  MOV    LUPKPT,R1          ;R1 AT NEXT PACKED ENTRY
6568  024070' 016702 000216                  MOV    LPHYSL,R2          ;R2 AT  PHYSICAL ADDRESS
6569  024074' 012605                         MOV    (SP)+,R5
6570  024076' 012604                         MOV    (SP)+,R4           ;RESTORE R3,R4,R5
6571  024100' 012603                         MOV    (SP)+,R3
6572  024102' 000205                         RTS    R5                 ;RETURN TO CALLING PGM
6573  024104' 005203                 LUPAMP: INC    R3                 ;ADVANCE MODEL POINTER
6574  024106' 122713 000377          LUPEMB: CMPB   #377,(R3)          ;EXAMINE MODEL BYTE
6575  024112' 001446                         BEQ    LUPVAR             ;GO UNPACK VARIABLE
6576  024114' 122713 000376                  CMPB   #376,(R3)
6577  024120' 001425                         BEQ    LUPLIN             ;GO UNPACK LINE NBR
6578  024122' 122713 000375                  CMPB   #375,(R3)
6579  024126' 001412                         BEQ    LUPBIT             ;GO UNPACK BIT NBRS
6580  024130' 122713 000374                  CMPB   #374,(R3)
6581  024134' 001460                         BEQ    LUPIMP             ;GO UNPACK IMM PRINT
6582  024136' 122713 000373                  CMPB   #373,(R3)
6583  024142' 001727                         BEQ    LUPEXT             ;TAKE SPECIAL EXIT
6584  024144' 105713                         TSTB   (R3)               ;IF MODEL BYTE=0
6585  024146' 001725                         BEQ    LUPEXT             ;EXIT
6586  024150' 111325                         MOVB   (R3),(R5)+         ;STORE MODEL CHARACTER
6587  024152' 000754                         BR     LUPAMP             ;THEN ADVANCE MODEL POINTER
6588  024154' 012400                 LUPBIT: MOV    (R4)+,R0
6589  024156' 112725 000040                  MOVB   #40,(R5)+
6590  024162' 010501                         MOV    R5,R1
6591  024164' 004567 001640                  JSR    R5,BTUPAK          ;UNPACK BIT NUMBERS
6592  024170' 010105                         MOV    R1,R5
6593  024172' 000744                         BR     LUPAMP             ;THEN ADVANCE MODEL POINTER
6594  024174' 005204                 LUPLIN: INC    R4                 ;BUMP R4 PAST 'L'
6595  024176' 112725 000040                  MOVB   #40,(R5)+          ;STORE A BLANK
6596  024202' 000401                         BR     LUPTG1
6597  024204' 112425                 LUPCTA: MOVB   (R4)+,(R5)+        ;STORE FIRST BYTE
6598  024206' 010400                 LUPTG1: MOV    R4,R0
6599  024210' 010501                         MOV    R5,R1
6600  024212' 004567 014606                  JSR    R5,DECASC          ;CONVERT LINE # TO ASCII
6601  024216' 010004                         MOV    R0,R4
6602  024220' 010105                         MOV    R1,R5              ;RESTORE POINTERS
6603  024222' 112725 000040                  MOVB   #40,(R5)+          ;STORE A BLANK
6604  024226' 000726                         BR     LUPAMP             ;THEN ADVANCE MODEL POINTER
6605  024230' 112725 000040          LUPVAR: MOVB   #40,(R5)+          ;STORE A BLANK
6606  024234' 121427 000040                  CMPB   (R4),#40           ;FIRST CHAR A BLANK ?
6607  024240' 001761                         BEQ    LUPCTA             ;YES-UNPACK OCTAL NBR
6608  024242' 121427 000104                  CMPB   (R4),#104          ;FIRST CHAR A "D" ?
6609  024246' 001004                         BNE    LUPTAG             ;NO-GO UNPACK TAG
6610  024250' 126427 000001 000360           CMPB   1(R4),#360         ;IF 2ND CHAR > 360
6611  024256' 101352                         BHI    LUPCTA             ;GO UNPACK DECIMAL NBR
6612  024260' 112425                 LUPTAG: MOVB   (R4)+,(R5)+
```

```
6613  024262' 112425                    MOVB    (R4)+,(R5)+
6614  024264' 112425                    MOVB    (R4)+,(R5)+
6615  024266' 112425                    MOVB    (R4)+,(R5)+         ;MOVE TAG AS IS
6616  024270' 112725  000040            MOVB    #40,(R5)+           ;STORE A BLANK
6617  024274' 000703                    BR      LUPAMP              ;THEN ADVANCE MODEL POINTER
6618  024276' 112425            LUPIMP: MOVB    (R4)+,(R5)+         ;MOVE STORED BYTE
6619  024300' 121427  000015            CMPB    (R4),#15            ;UNTIL CARRIAGE RET
6620  024304' 001374                    BNE     LUPIMP
6621  024306' 000676                    BR      LUPAMP              ;THEN ADVANCE MODEL PTR
6622
6623  024310'     000           LENDFL: .BYTE   0
6624  024311'     000           LPLNTH: .BYTE   0
6625  024312' 000000            LPHYSL: .WORD   0
6626  024314' 000000            LUPKPT: .WORD   0
6627  024316' 000000            LUPPTP: .WORD   0
```

```
                                        .SBTTL  SYNTAX ERROR SUBROUTINE

                                ;*****************************************************************
                                ;
                                ;       THIS SUBROUTINE PRINTS A LINE UNDERNEATH THE USERS
                                ;       ENTRY, WHEN ANYTHING IN THAT ENTRY CANNOT BE INTERPRETED.
                                ;       THIS LINE CONSISTS OF ALL BLANKS EXCEPT FOR THE BAD
                                ;       ENTRY SURROUNDED BY QUESTION MARKS
                                ;
                                ;       CALLING SEQUENCE        JSR     R5,SYNTAX
                                ;
                                ;*****************************************************************

024320' 010700                  SYNTAX: MOV     PC,R0
024322' 062700  007740                  ADD     #LINBUF-.,R0            ;SET PNTR TO LINE BUFF
024326' 016701  002274                  MOV     LSCNCT,R1               ;GET CHAR COUNT AND
024332' 016702  002266                  MOV     LSCNPT,R2               ;LAST POS OF BAD WORD
024336' 012767  000005  000114          MOV     #5,LTTYCT               ;INIT OUTPUT COUNT
024344' 005202                          INC     R2                      ;POINT R2 PAST BAD WORD
024346' 112722  000077                  MOVB    #'?,(R2)+               ;STORE A ?
024352' 112722  000015                  MOVB    #15,(R2)+               ;A CARRIAGE RETURN
024356' 112712  000012                  MOVB    #12,(R2)                ;AND A LINE FEED
024362' 060167  000072                  ADD     R1,LTTYCT               ;ADJUST OUTPUT COUNT
024366' 162702  000003                  SUB     #3,R2
024372' 160102                          SUB     R1,R2                   ;POINT R2 BEFORE BAD WORD
024374' 111262  177777                  MOVB    (R2),-1(R2)             ;MOVE THAT CHAR 1 BACK
024400' 112712  000077                  MOVB    #'?,(R2)                ;INSERT A ?
024404' 020200                  LSNLP1: CMP     R2,R0                   ;ARE WE AT LINE START ?
024406' 001414                          BEQ     LSNTG2                  ;YES - GET READY TO TYPE
024410' 122762  000011  177777          CMPB    #11,-1(R2)              ;IS THIS A TAB ?
024416' 001003                          BNE     LSNTG1                  ;NO - CONTINUE
024420' 005302                          DEC     R2                      ;YES - SKIP IT
024422' 005267  000032                  INC     LTTYCT                  ;AND BUMP OUTPUT COUNT
024426' 112742  000040          LSNTG1: MOVB    #40,-(R2)               ;INSERT A BLANK
024432' 005267  000022                  INC     LTTYCT                  ;BUMP OUTPUT COUNT
024436' 000762                          BR      LSNLP1                  ;CHECK IF DONE
024440' 012722  051105          LSNTG2: MOV     #"ER,(R2)+              ;STORE 'ER'
024444' 112712  000052                  MOVB    #'*,(R2)                ;THEN ANOTHER *
024450' 004567  161550                  JSR     R5,CTYPE                ;TYPE THE LINE
024454' 000420                          .WORD   420
024456' 007603                          .WORD   LINBUF-1-.
024460' 000001                  LTTYCT: .WORD   1
024462' 105267  176563                  INCB    LERROR                  ;SET ERROR FLAG
024466' 000205                          RTS     R5                      ;RETURN TO CALLING PGM.
```

```
                                .SBTTL  DISPLAY COMMAND HANDLER

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE HANDLES THE DISPLAY N
                        ;       AND DISPLAY N CODE COMMANDS.  THE OUTPUT
                        ;       IS ON THE DISPLAY DEVICE AND CONSISTS
                        ;       OF A LISTING OF THE USERS STATEMENTS
                        ;       ALONG WITH THE MACHINE CODE GENERATED
                        ;       BY THE COMPILER (IF REQUESTED).
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,DISP (DISPCD)
                        ;
                        ;       ENTRY   R3 POINTS TO PROGRAM TABLE
                        ;
                        ;*****************************************************************

024470' 105067  000164          DISP:   CLRB    LSTOBJ                  ;CLEAR LIST OBJ FLAG
024474' 000402                  BR      LDISVR                  ;THEN SAVE REGISTERS
024476' 105267  000156          DISPCD: INCB    LSTOBJ                  ;SET LIST OBJ FLAG
024502' 010367  177610          LDISVR: MOV     R3,LUPPTP               ;SAVE P TBL POINTER
024506' 010346                  MOV     R3,-(SP)
024510' 010446                  MOV     R4,-(SP)                ;SAVE R3 AND R4
024512' 016367  000022  177574  MOV     PSRCST(R3),LUPKPT       ;INIT UNPACK POINTER
                        001     .IF DF MM
                                BIS     #P4CONS,LUPKPT          ;SET PAR4 SELECT BITS
                        000     .ENDC
024520' 004567  176650          LDILP1: JSR     R5,UNPACK               ;UNPACK ONE LINE
024524' 010067  000010          MOV     R0,LDITBC
024530' 010204                  MOV     R2,R4                   ;SAVE PHYSICAL ADDRESS
                        001     .IF DF MM
                                BIS     #P4CONS,R4              ;SET PAR4 SELECT BITS
                        000     .ENDC
024532' 004567  011042          JSR     R5,CLIST                ;PRINT IT
024536' 007524                  .WORD   LINBUF-.
024540' 000000          LDITBC: .WORD   0
024542' 105767  000112          TSTB    LSTOBJ                  ;TEST LIST OBJ FLAG
024546' 001006                  BNE     LDIGPA                  ;BRANCH IF SET
024550' 105767  177534  LDITEF: TSTB    LENDFL                  ;TEST END FLAG
024554' 001761                  BEQ     LDILP1                  ;LOOP IF NOT SET
024556' 012604                  MOV     (SP)+,R4
024560' 012603                  MOV     (SP)+,R3                ;RESTORE REGISTERS
024562' 000205                  RTS     R5                      ;RETURN TO CALLING PGM
                        001     .IF DF MM
                                .IFF
024564' 005704          LDIGPA: TST     R4                      ;IF PHYSICAL ADDR ZERO
                                .IFT
                        LDIGPA: CMP     R4,#P4CONS              ;IF PHYSICAL ADR ZERO
                        000     .ENDC
024566' 001770                  BEQ     LDITEF                  ;GO CHECK END FLAG
024570' 105767  177514          TSTB    LENDFL                  ;DETOUR
024574' 001025                  BNE     LDIREF                  ;IF END FLAG SET
024576' 016702  177512          MOV     LUPKPT,R2
024602' 016203  000004          MOV     4(R2),R3                ;GET NEXT PHYSICAL ADDR
024606' 001760                  BEQ     LDITEF                  ;IF ZERO DONT CONTINUE
```

```
6731                    001                         .IF DF MM
6732                                                .IFT
6733                                        LDILCD: MOV     @#KPAR4,(PC)+           ;GET 22 BIT ADDRESS
6734                                        LPARSV: .WORD   0
6735                                                JSR     R5,G22BAD               ;OF OBJ CODE WORD
6736                                                .WORD   LPARSV-.
6737                                                MOV     R4,-(SP)
6738                                                SUB     #P4CONS,(SP)
6739                                                ADD     (SP)+,R0                ;PUT PHYSICAL ADR IN
6740                                                ADC     R2                      ;R2 AND R0
6741                                                JSR     R5,BINAS8               ;CONVERT TO ASCII
6742                                                .WORD   LDI18B-.
6743                                                MOV     R4,R0                   ;GET VIRTUAL ADR OF OBJ CODE WD
6744                                                BIC     #P4CONS,R0              ;RESET PAGE SELECT BITS
6745                                                .IFF
6746  024610' 010400                        LDILCD: MOV     R4,R0                   ;GET ADR OF OBJ CODE WORD
6747                    000                         .ENDC
6748  024612' 004567  014304                        JSR     R5,BINASC               ;CONVERT OBJ ADR
6749  024616' 000044                                .WORD   LDIADR-.                ;TO ASCII
6750  024620' 012400                                MOV     (R4)+,R0
6751  024622' 004567  014274                        JSR     R5,BINASC               ;CONVERT CONTENTS TO ASCII
6752  024626' 000044                                .WORD   LDIOBJ-.
6753  024630' 004567  010744                        JSR     R5,CLIST                ;PRINT OUT ABOVE
6754  024634' 000025                                .WORD   LDIMSG-.
6755                    001                         .IF DF MM
6756                                                .IFF
6757  024636' 000021                                .WORD   17.
6758                                                .IFT
6759                                                .WORD   29.
6760                    000                         .ENDC
6761  024640' 010400                                MOV     R4,R0
6762                    001                         .IF DF MM
6763                                                BIC     #P4CONS,R0              ;CLEAR PAR4 SELECT BITS
6764                    000                         .ENDC
6765  024642' 020003                                CMP     R0,R3                   ;CURRENT ADDR >=NEXT
6766  024644' 103761                                BLO     LDILCD                  ;NO - LIST MORE CODE
6767  024646' 000740                                BR      LDITEF                  ;YES - CHECK FOR MORE
6768  024650' 010403                        LDIREF: MOV     R4,R3                   ;GET CURRENT PHYSICAL ADDR
6769                    001                         .IF DF MM
6770                                                BIC     #P4CONS,R3              ;RESET PAR4 SELECT BITS
6771                    000                         .ENDC
6772  024652' 062703  000004                        ADD     #4,R3                   ;ADD 4 STORE AS NEXT
6773  024656' 000754                                BR      LDILCD                  ;GO PRINT CODE
6774
6775
6776  024660'     000                       LSTOBJ: .BYTE   0                       ;LIST OBJECT FLAG
6777
6778  024661'     011                       LDIMSG: .ASCII  <011>
6779                    001                         .IF DF MM
6780                                                .IFF
6781  024662' 054130  054130  054130        LDIADR: .ASCII  /XXXXXX= /
      024670' 020075
6782                                                .IFT
6783                                        LDI18B: .ASCII  /XXXXXXXX (/
6784                                        LDIADR: .ASCII  /XXXXXX) = /
6785                    000                         .ENDC
```

```
 6786  024672' 054130  054130  054130  LDIOBJ: .ASCII  /XXXXXX/<015><012>
       024700' 005015
 6787                                          .EVEN
```

```
                                .SBTTL  SAVE COMMAND HANDLER

                        ;*******************************************************************
                        ;
                        ;       THIS SUBROUTINE HANDLES THE SAVE COMMAND.
                        ;       IT DETERMINES THE NUMBER OF BLOCKS REQUIRED
                        ;       AND OPENS THE FILE.  IT UNPACKS THE INTERNAL
                        ;       PACKED CODE BACK TO ITS ORIGINAL FORM AND
                        ;       STORES IT ON THE LIBRARY DEVICE.
                        ;
                        ;       CALLING SEQUENCE JSR    R5,SAVE
                        ;
                        ;       ENTRY   R3 POINTS TO PROGRAM TABLE
                        ;
                        ;*******************************************************************

024702' 005067  000402          SAVE:   CLR     LSVFPS                  ;CLEAR FIRST PASS SW
024706' 010367  177404                  MOV     R3,LUPPTP               ;SAVE P TBL POINTER
024712' 016367  000022  177374  LSVPS2: MOV     PSRCST(R3),LUPKPT       ;INIT UNPACK POINTER
            001                 .IF DF MM
                                        BIS     #P4CONS,LUPKPT          ;SET PAR4 SELECT BITS
            000                 .ENDC
024720' 005067  000366                  CLR     LSVRTL                  ;CLEAR RUNNING TOTAL
024724' 012767  000754  000354          MOV     #492.,LSVBCT            ;SET BYTES REMAIN=492
024732' 016704  153052                  MOV     BUF,R4                  ;GET BLOCK BUF POINTER
024736' 062704  000010                  ADD     #8.,R4                  ;SKIP FIRST 8 BYTES
024742' 005067  000174                  CLR     LSVBNM                  ;SET NBR BLKS=0
024746' 005767  000336                  TST     LSVFPS                  ;TEST FIRST PASS SW
024752' 001003                          BNE     LSVRSW                  ;IF SET-RESET IT
024754' 005267  000330                  INC     LSVFPS                  ;OTHERWISE SET IT
024760' 000402                          BR      LSVUPL
024762' 005067  000322  LSVRSW: CLR     LSVFPS                  ;RESET FIRST PASS SW
024766' 004567  176402  LSVUPL: JSR     R5,UNPACK               ;UNPACK ONE LINE
024772' 010702                  MOV     PC,R2
024774' 062702  007266          ADD     #LINBUF-.,R2            ;POINTER R2 TO LINBUF

025000' 160067  000302          SUB     R0,LSVBCT               ;SUBTRACT CHAR COUNT
025004' 100461                  BMI     LSVWFT                  ;BRANCH IF IT WONT FIT
025006' 112224  LSVMLB: MOVB    (R2)+,(R4)+             ;MOVE LINE TO BLOCK
025010' 005300                  DEC     R0
025012' 001375                  BNE     LSVMLB
025014' 105767  177270  LSVTEN: TSTB    LENDFL                  ;TEST END FLAG
025020' 001762                  BEQ     LSVUPL                  ;IF RESET UNPACK MORE
025022' 010702                  MOV     PC,R2
025024' 062702  007236          ADD     #LINBUF-.,R2
025030' 012722  047504          MOV     #"DO,(R2)+              ;MOVE 'DONE'
025034' 012722  042516          MOV     #"NE,(R2)+              ;TO LINE BUFFER
025040' 016722  007364          MOV     LCRLF,(R2)+
025044' 012700  000006          MOV     #6,R0
025050' 160002                  SUB     R0,R2
025052' 160067  000230          SUB     R0,LSVBCT               ;SUBTRACT CHAR CNT
025056' 100434                  BMI     LSVWFT                  ;BRANCH IF IT WONT FIT
025060' 112224  LSVMDB: MOVB    (R2)+,(R4)+             ;MOVE 'DONE' TO BLOCK
025062' 005300                  DEC     R0
025064' 001375                  BNE     LSVMDB
```

```
 6845  025066' 005767  000216                TST     LSVFPS          ;TEST FIRST PASS SW
 6846  025072' 001017                          BNE     LSVBBC          ;IF SET BUMP BLK CNT
 6847  025074' 016704  152710                  MOV     BUF,R4          ;GET BLOCK BUF POINTER
 6848  025100' 062704  000004                  ADD     #4,R4           ;BUMP IT BY 4
 6849  025104' 012701  000762                  MOV     #498.,R1
 6850  025110' 166701  000172                  SUB     LSVBCT,R1       ;CALC BYTES + 6
 6851  025114' 010124                          MOV     R1,(R4)+        ;STORE IN BUFFER
 6852  025116' 016714  000170                  MOV     LSVRTL,(R4)     ;ALSO STORE RUNNING TOTAL
 6853  025122' 004577  152670                  JSR     R5,@CLOSE       ;IF NOT CLOSE FILE
 6854  025126' 000002                          .WORD   LSVEXT-.
 6855  025130' 000205          LSVEXT: RTS     R5                      ;RETURN TO CONTROL ROUT.
 6856  025132' 005267  000004  LSVBBC: INC     LSVBNM                  ;BUMP BLOCK COUNTER
 6857  025136' 004577  152660          JSR     R5,@CREATE              ;OPEN A FILE
 6858  025142' 000000          LSVBNM: .WORD   0                       ;WITH THIS NBR BLKS
 6859  025144' 177764                          .WORD   LSVEXT-.
 6860  025146' 000661                          BR      LSVPS2          ;GO TO SECOND PASS
 6861  025150' 060067  000132  LSVWFT: ADD     R0,LSVBCT               ;ADD CHAR COUNT BACK
 6862  025154' 005767  000126                  TST     LSVBCT          ;REMAINING BYTE COUNT = 0 ?
 6863  025160' 001405                          BEQ     LSVBCZ          ;YES GOTO LSVBCZ
 6864  025162' 112224          LSVMOC: MOVB    (R2)+,(R4)+             ;MOVE ONE CHARACTER
 6865  025164' 005300                          DEC     R0              ;AT A TIME
 6866  025166' 005367  000114                  DEC     LSVBCT
 6867  025172' 001373                          BNE     LSVMOC
 6868  025174' 005767  000110  LSVBCZ: TST     LSVFPS                  ;TEST FIRST PASS SW
 6869  025200' 001022                          BNE     LSVDWR          ;DONT WRITE IF SET
 6870  025202' 016704  152602                  MOV     BUF,R4          ;GET BLOCK BUF POINTER
 6871  025206' 062704  000004                  ADD     #4,R4           ;BUMP IT BY 4
 6872  025212' 012701  000762                  MOV     #498.,R1
 6873  025216' 166701  000064                  SUB     LSVBCT,R1       ;CALC BYTES + 6
 6874  025222' 010124                          MOV     R1,(R4)+        ;STORE  IN BUFFER
 6875  025224' 016714  000062                  MOV     LSVRTL,(R4)     ;ALSO STORE RUNNING TOTAL
 6876  025230' 162701  000006                  SUB     #6,R1
 6877  025234' 060167  000052                  ADD     R1,LSVRTL       ;UPDATE RUNNING  TOTAL
 6878  025240' 004577  152612                  JSR     R5,@PUT         ;WRITE ONE BLOCK
 6879  025244' 177664                          .WORD   LSVEXT-.
 6880  025246' 005267  177670  LSVDWR: INC     LSVBNM                  ;BUMP BLOCK COUNTER
 6881  025252' 016704  152532                  MOV     BUF,R4          ;GET BLOCK BUF POINTER
 6882  025256' 012767  000754  000022          MOV     #492.,LSVBCT    ;SET BYTES REMAIN=492
 6883  025264' 062704  000010                  ADD     #8.,R4          ;SKIP FIRST 8 BYTES
 6884  025270' 112224          LSVMRL: MOVB    (R2)+,(R4)+             ;MOVE REST OF LINE
 6885  025272' 005367  000010                  DEC     LSVBCT
 6886  025276' 005300                          DEC     R0
 6887  025300' 001373                          BNE     LSVMRL
 6888  025302' 000167  177506                  JMP     LSVTEN          ;THEN GO TEST END FLAG
 6889
 6890  025306' 000000          LSVBCT: .WORD   0                       ;REMAINING BYTE COUNT
 6891  025310' 000000          LSVFPS: .WORD   0                       ;FIRST PASS SWITCH
 6892  025312' 000000          LSVRTL: .WORD   0                       ;RUNNING TOT - BYTES WRITTEN
```

```
                        .SBTTL  BIT PACKER SUBROUTINE

                ;*****************************************************************
                ;
                ;       THIS SUBROUTINE INTERPRETS THE USER STATEMENT
                ;       WORDS N AND N THRU N OR N AND N -N AND SETS
                ;       THE APPROPRIATE BITS IN A 16 BIT WORD.
                ;
                ;       CALLING SEQUENCE        JSR     R5,BTPACK
                ;                               .WORD   ERRTN-.
                ;
                ;       ENTRY   R0 CONTAINS THE NUMBER OF USER WORDS IN THE
                ;                       STRING (EXAMPLE -7 FOR N & N & N - N)
                ;               LSCNPT POINTS TO THE WORD BEFORE THE STRING
                ;
                ;       EXIT    R0 CONTAINS THE 16 BIT WORD WITH BITS SET
                ;
                ;*****************************************************************

025314' 010067 000460          BTPACK: MOV     R0,LNBRWD               ;SAVE NBR OF WORDS
025320' 012767 000001 000450           MOV     #1,LBTPNT               ;SET BIT POINTER TO BIT 0
025326' 004567 001054                  JSR     R5,SCAN                 ;FIND FIRST WORD
025332' 004567 015012                  JSR     R5,ASDECO               ;CONVERT TO BCD
025336' 000434                         .WORD   LBPERX-.
025340' 004567 014552                  JSR     R5,DECBIN               ;CONVERT BCD TO BINARY
025344' 000426                         .WORD   LBPERX-.
025346' 010067 000430                  MOV     R0,LBTNBR               ;STORE RESULT AS BIT NBR
025352' 010067 000430                  MOV     R0,LBTNSV               ;AND SAVE FOR LATER
025356' 005367 000416                  DEC     LNBRWD                  ;DECREMENT NUMBER OF WORDS
025362' 005067 000416                  CLR     LBPRWD                  ;SET RESULT WORD TO ZEROES
025366' 005767 000410                  TST     LBTNBR                  ;IF REQUESTED BIT NBR 0
025372' 001410                         BEQ     LBPBS                   ;GO SET THAT BIT
025374' 006367 000376          LBPLP1: ASL     LBTPNT                  ;SHIFT BIT POINTER LEFT
025400' 001002                         BNE     LBPTG1                  ;ERROR IF SHIFT >16
025402' 000167 000364                  JMP     LBPERX
025406' 005367 000370          LBPTG1: DEC     LBTNBR                  ;DECREMENT BIT NUMBER
025412' 001370                         BNE     LBPLP1                  ;SHIFT TO CORRECT POSITION
025414' 056767 000356 000362   LBPBS:  BIS     LBTPNT,LBPRWD           ;SET A BIT
025422' 005767 000352          LBPMOR: TST     LNBRWD
025426' 001555                         BEQ     LBPEXT                  ;EXIT IF ALL WORDS PROCESSED
025430' 004567 000752                  JSR     R5,SCAN
025434' 005301                         DEC     R1
025436' 160100                         SUB     R1,R0                   ;MOVE POINTER TO WORD START
025440' 010702                         MOV     PC,R2
025442' 062702 000346                  ADD     #LBPVTB-.,R2            ;POINT R2 AT LOOKUP TABLE
025446' 005367 000326                  DEC     LNBRWD                  ;DECREMENT NUMBER OF WORDS
025452' 001547                         BEQ     LBPERX                  ;ERROR IF EVEN # WORDS
025454' 012701 000004                  MOV     #4,R1
025460' 121012                 LBPLP2: CMPB    (R0),(R2)               ;IS WORD A TABLE ENTRY
025462' 001003                         BNE     LBPNF                   ;BRANCH IF NO
025464' 062702 000002                  ADD     #2,R2
025470' 011207                         MOV     (R2),PC                 ;IF YES TAKE BRANCH
025472' 062702 000004          LBPNF:  ADD     #4,R2                   ;BUMP TABLE POINTER
025476' 005301                         DEC     R1                      ;END OF TABLE
```

```
6950  025500' 001367                      BNE     LBPLP2          ;NO-CHECK NEXT ENTRY
6951  025502' 000533                      BR      LBPERX          ;YES-TAKE ERROR EXIT
6952  025504' 004567  000676      LBPAND: JSR     R5,SCAN         ;FIND NEXT WORD
6953  025510' 005367  000264              DEC     LNBRWD          ;DECREMENT NBR OF WORDS
6954  025514' 004567  014630              JSR     R5,ASDECO       ;CONVERT WORD (BIT#) TO BCD
6955  025520' 000252                      .WORD   LBPERX-.
6956  025522' 004567  014370              JSR     R5,DECBIN       ;CONVERT BCD TO BINARY
6957  025526' 000244                      .WORD   LBPERX-.
6958  025530' 010067  000246              MOV     R0,LBTNBR       ;STORE RESULT AS BIT NBR
6959  025534' 010067  000246              MOV     R0,LBTNSV       ;AND SAVE FOR LATER
6960  025540' 012767  000001  000230      MOV     #1,LBTPNT       ;SET BIT POINTER TO BIT 0
6961  025546' 005767  000230              TST     LBTNBR          ;IF BIT NUMBER 0
6962  025552' 001406                      BEQ     LBPSBO          ;GO SET IT
6963  025554' 006367  000216      LBPLP3: ASL     LBTPNT          ;SHIFT BIT POINTER LEFT
6964  025560' 001504                      BEQ     LBPERX          ;ERROR IF SHIFT >16
6965  025562' 005367  000214              DEC     LBTNBR          ;DECREMENT BIT NBR
6966  025566' 001372                      BNE     LBPLP3          ;SHIFT TO CORRECT POSITION
6967  025570' 056767  000202  000206 LBPSBO: BIS  LBTPNT,LBPRWD   ;SET A BIT
6968  025576' 005767  000176              TST     LNBRWD          ;TEST IF NBR OF WORDS =0
6969  025602' 001467                      BEQ     LBPEXT          ;EXIT IF YES
6970  025604' 000706                      BR      LBPMOR          ;IF NO TEST MORE WORDS
6971  025606' 004567  000574      LBPTRU: JSR     R5,SCAN         ;FIND NEXT WORD
6972  025612' 005367  000162              DEC     LNBRWD          ;DECREMENT NBR OF WORDS
6973  025616' 004567  014526              JSR     R5,ASDECO       ;CONVERT WORD (BIT#) TO BCD
6974  025622' 000150                      .WORD   LBPERX-.
6975  025624' 004567  014266              JSR     R5,DECBIN       ;CONVERT BCD TO BINARY
6976  025630' 000142                      .WORD   LBPERX-.
6977  025632' 016701  000150              MOV     LBTNSV,R1       ;GET PRIOR BIT#
6978  025636' 010067  000144              MOV     R0,LBTNSV       ;SAVE LATEST
6979  025642' 020001                      CMP     R0,R1
6980  025644' 002403                      BLT     LBPPGL          ;IF LATEST >PRIOR
6981  025646' 010067  000130              MOV     R0,LBTNBR       ;STORE LATEST
6982  025652' 000403                      BR      LBPLGP          ;IF PRIOR >LATEST
6983  025654' 010167  000122      LBPPGL: MOV     R1,LBTNBR       ;STORE PRIOR
6984  025660' 010001                      MOV     R0,R1           ;LATEST TO R1
6985  025662' 012767  000001  000106 LBPLGP: MOV  #1,LBTPNT       ;SET BIT POINTER TO BIT 0
6986  025670' 005701                      TST     R1              ;IF BIT # IS NOT 0
6987  025672' 001001                      BNE     LBPLP4          ;CONTINUE
6988  025674' 000407                      BR      LBPTG2          ;OTHERWISE SET BIT 0
6989  025676' 006367  000074      LBPLP4: ASL     LBTPNT          ;SHIFT BIT POINTER LEFT
6990  025702' 001433                      BEQ     LBPERX          ;ERROR IF SHIFT >16
6991  025704' 005367  000072              DEC     LBTNBR          ;DECREMENT LARGER BIT#
6992  025710' 005301                      DEC     R1              ;DECREMENT SMALLER BIT#
6993  025712' 001371                      BNE     LBPLP4          ;SHIFT TO CORRECT POSITION
6994  025714' 020167  000062      LBP     CMP     R1,LBTNBR       ;CHECK IF PRIOR =LATEST
6995  025720' 001414                      BEQ     LBPST1          ;GO SET ONE BIT
6996  025722' 056767  000050  000054 LBPLP5: BIS  LBTPNT,LBPRWD   ;SET A BIT
6997  025730' 006367  000042              ASL     LBTPNT          ;SHIFT BIT POINTER LEFT
6998  025734' 001001                      BNE     LBPTG3
6999  025736' 103015                      BCC     LBPERX          ;EXIT IF SHIFT > 16
7000  025740' 005201              LBPTG3: INC     R1              ;INCREMENT SMALLER NBR
7001  025742' 020167  000034              CMP     R1,LBTNBR       ;COMPARE TO LARGER
7002  025746' 101765                      BLOS    LBPLP5          ;LOOP IF < OR =
7003  025750' 000624                      BR      LBPMOR
7004  025752' 056767  000020  000024 LBPST1: BIS  LBTPNT,LBPRWD   ;SET ONE BIT
7005  025760' 000620                      BR      LBPMOR
```

```
 7006  025762' 016700  000016          LBPEXT: MOV     LBPRWD,R0               ;RESULT TO R0
 7007  025766' 005725                          TST     (R5)+                   ;BYPASS ERR RETURN ADR
 7008  025770' 000205                          RTS     R5                      ;RETURN TO CALLING PGM
 7009
 7010  025772' 061505                  LBPERX: ADD     (R5),R5                 ;CALCULATE ERROR RETURN
 7011  025774' 000205                          RTS     R5                      ;AND TAKE IT
 7012
 7013  025776' 000000                  LBTPNT: .WORD   0                       ;BIT POINTER
 7014  026000' 000000                  LNBRWD: .WORD   0                       ;NUMBER OF WORDS
 7015  026002' 000000                  LBTNBR: .WORD   0                       ;BIT NUMBER
 7016  026004' 000000                  LBPRWD: .WORD   0                       ;BIT PACKER RESULT
 7017  026006' 000000                  LBTNSV: .WORD   0                       ;BIT NUMBER SAVE
 7018
 7019  026010' 000101                  LBPVTB: .ASCIZ  /A/                     ;AND-THRU LOOKUP TABLE
 7020  026012' 025504'                 LBPRT1: .WORD   LBPAND
 7021  026014' 000046                          .ASCIZ  /&/
 7022  026016' 025504'                 LBPRT2: .WORD   LBPAND
 7023  026020' 000124                          .ASCIZ  /T/
 7024  026022' 025606'                 LBPRT3: .WORD   LBPTRU
 7025  026024' 000055                          .ASCIZ  /-/
 7026  026026' 025606'                 LBPRT4: .WORD   LBPTRU
```

```
                                .SBTTL  BIT UNPACKER SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE CONVERTS A 16 BIT
                        ;       BINARY NUMBER INTO THE FORM OF
                        ;       N THRU N AND N AS AN ASCII CHARACTER STRING
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,BTUPAK
                        ;
                        ;
                        ;       ENTRY   R0 CONTAINS THE 16 BIT NUMBER
                        ;               R1 POINTS TO WHERE ASCII STRING DESIRED
                        ;
                        ;       EXIT    R1 POINTS TWO SPACES BEYOND LAST
                        ;                       WORD GENERATED.
                        ;
                        ;*****************************************************************

026030' 010067 000310           BTUPAK: MOV     R0,LBUWRD               ;SAVE BINARY NUMBER
026034' 012767 000001 000262            MOV     #1,LBUBPT               ;SET BIT 0 OF UNPACK PTR
026042' 005067 000260                   CLR     LBUNBR                  ;SET BIT NBR TO 0
026046' 036767 000252 000270    LBULP1: BIT     LBUBPT,LBUWRD           ;TEST IF BIT SET
026054' 001011                          BNE     LBUSAS                  ;BRANCH IF IT IS
026056' 005267 000244                   INC     LBUNBR                  ;INCR BIT NBR
026062' 006367 000236                   ASL     LBUBPT                  ;SHIFT BIT POINTER LEFT
026066' 100367                          BPL     LBULP1                  ;REPEAT IF NOT BIT 15
026070' 036767 000230 000246            BIT     LBUBPT,LBUWRD           ;TEST IF BIT 15 SET
026076' 001504                          BEQ     LBUEXT                  ;EXIT IF NOT
026100' 016702 000222           LBUSAS: MOV     LBUNBR,R2               ;GET BIT NUMBER
026104' 006302                          ASL     R2                      ;DOUBLE IT
026106' 060702                          ADD     PC,R2                   ;CALCULATE
026110' 062702 000236                   ADD     #LBUAST-.,R2            ;LOOKUP TABLE POSITION
026114' 112221                          MOVB    (R2)+,(R1)+             ;PRINT FIRST DIGIT
026116' 111221                          MOVB    (R2),(R1)+              ;THEN SECOND
026120' 022767 000017 000200            CMP     #15.,LBUNBR             ;TEST IF LAST BIT # 15
026126' 001470                          BEQ     LBUEXT                  ;EXIT IF YES
026130' 006367 000170                   ASL     LBUBPT                  ;ADVANCE BIT POINTER
026134' 005267 000166                   INC     LBUNBR                  ;INCREMENT BIT NUMBER
026140' 036767 000160 000176            BIT     LBUBPT,LBUWRD           ;TEST IF BIT SET
026146' 001433                          BEQ     LBUFBT                  ;BRANCH IF NOT
026150' 012700 000006                   MOV     #6,R0
026154' 010702                          MOV     PC,R2
026156' 062702 000152                   ADD     #LBUTRU-.,R2
026162' 112221                  LBULP2: MOVB    (R2)+,(R1)+             ;STORE WORD 'THRU'
026164' 005300                          DEC     R0
026166' 001375                          BNE     LBULP2
026170' 006367 000130           LBULP3: ASL     LBUBPT                  ;ADVANCE BIT POINTER
026174' 005267 000126                   INC     LBUNBR                  ;INCREMENT BIT NUMBER
026200' 036767 000120 000136            BIT     LBUBPT,LBUWRD           ;TEST IF BIT SET
026206' 001405                          BEQ     LBUBKU                  ;BRANCH IF NOT
026210' 022767 000017 000110            CMP     #15.,LBUNBR             ;TEST IF BIT 15
026216' 101364                          BHI     LBULP3                  ;LOOP IF NOT
026220' 000727                          BR      LBUSAS                  ;STORE 15 IF IT IS
026222' 000241                  LBUBKU: CLC
```

```
7084  026224' 006067 000074                RUR     LBUBPT                  ;BACK UP POINTER
7085  026230' 005367 000072                DEC     LBUNBR                  ;DECR BIT #
7086  026234' 000721                       BR      LBUSAS                  ;THEN STORE NUMBER
7087  026236' 022767 000017 000062 LBUFBT: CMP     #15.,LBUNBR
7088  026244' 001421                       BEQ     LBUEXT                  ;EXIT IF BIT # = 15
7089  026246' 006367 000052                ASL     LBUBPT                  ;ADVANCE BIT POINTER
7090  026252' 005267 000050                INC     LBUNBR
7091  026256' 036767 000042 000060        BIT     LBUBPT,LBUWRD           ;TEST IF BIT SET
7092  026264' 001764                       BEQ     LBUFBT                  ;LOOP IF NOT
7093  026266' 012700 000005                MOV     #5,R0
7094  026272' 010702                       MOV     PC,R2
7095  026274' 062702 000042                ADD     #LBUAND-.,R2
7096  026300' 112221          LBULP4:      MOVB    (R2)+,(R1)+             ;STORE WORD "AND"
7097  026302' 005300                       DEC     R0
7098  026304' 001375                       BNE     LBULP4
7099  026306' 000674                       BR      LBUSAS                  ;STORE BIT NUMBER
7100  026310' 010702          LBUEXT:      MOV     PC,R2
7101  026312' 062702 000030                ADD     #LBUSPC-.,R2
7102  026316' 112221                       MOVB    (R2)+,(R1)+             ;STORE TWO SPACES
7103  026320' 111211                       MOVB    (R2),(R1)
7104  026322' 000205                       RTS     R5                      ;RETURN TO CALLING PGM
7105
7106  026324' 000000          LBUBPT:      .WORD   0                       ;UNPACK BIT POINTER
7107  026326' 000000          LBUNBR:      .WORD   0                       ;UNPACK BIT NUMBER
7108  026330' 052040 051110 020125 LBUTRU: .ASCII  / THRU /
7109  026336' 040440 042116        LBUAND: .ASCII  / AND/
7110  026342' 020040                LBUSPC: .ASCII  /  /
7111  026344' 000000                LBUWRD: .WORD   0
7112  026346' 030040                LBUAST: .ASCII  / 0/
7113  026350' 030440                        .ASCII  / 1/
7114  026352' 031040                        .ASCII  / 2/
7115  026354' 031440                        .ASCII  / 3/
7116  026356' 032040                        .ASCII  / 4/
7117  026360' 032440                        .ASCII  / 5/
7118  026362' 033040                        .ASCII  / 6/
7119  026364' 033440                        .ASCII  / 7/
7120  026366' 034040                        .ASCII  / 8/
7121  026370' 034440                        .ASCII  / 9/
7122  026372' 030061                        .ASCII  /10/
7123  026374' 030461                        .ASCII  /11/
7124  026376' 031061                        .ASCII  /12/
7125  026400' 031461                        .ASCII  /13/
7126  026402' 032061                        .ASCII  /14/
7127  026404' 032461                        .ASCII  /15/
```

```
                                .SBTTL  SCAN SUBROUTINE

                        ;*************************************************************
                        ;
                        ;
                        ;       THIS SUBROUTINE SCANS A LINE OF ASCII CHARACTERS
                        ;       AND EXITS WHEN THE NEXT WORD OR SPECIAL SYMBOL
                        ;       IS FOUND.  SCANNING BEGINS AT THE ADDRESS STORED
                        ;       IN LSCNPT.  ANY INITIAL BLANKS ARE IGNORED.
                        ;       WHEN THE FIRST NON-BLANK CHARACTER IS ENCOUNTERED
                        ;       SCAN BEGINS COUNTING.  WHEN THE NEXT BLANK, DASH, COMMA,
                        ;       TAB, AMPERSAND OR CARRIAGE RETURN IS ENCOUNTERED,
                        ;       SCAN EXITS.  R0 AND LSCNPT THEN CONTAIN THE ADDRESS OF
                        ;       THE LAST CHARACTER OF THE WORD, AND R1 CONTAINS A COUNT
                        ;       OFTHE NUMBER OF CHARACTERS.
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,SCAN
                        ;*************************************************************

026406' 016700 000212   SCAN:   MOV     LSCNPT,R0       ;GET SCAN POINTER
026412' 005200                  INC     R0              ;ADVANCE IT BY ONE
026414' 005001                  CLR     R1              ;CLEAR CHARACTER COUNT
026416' 112002          LSKBL:  MOVB    (R0)+,R2        ;GET CHAR, ADVANCE POINTER
026420' 122702 000015           CMPB    #15,R2          ;CARRIAGE RETURN ?
026424' 001454                  BEQ     LSCOUT          ;YES - GET OUT
026426' 122702 000055           CMPB    #55,R2          ;HYPHEN?
026432' 001451                  BEQ     LSCOUT          ;YES - GET OUT
026434' 122702 000054           CMPB    #54,R2          ;COMMA ?
026440' 001446                  BEQ     LSCOUT          ;YES - GET OUT
026442' 122702 000052           CMPB    #'*,R2          ;ASTERISK ?
026446' 001443                  BEQ     LSCOUT          ;YES - GET OUT
026450' 122702 000043           CMPB    #'#,R2          ;NBR SGN ?
026454' 001440                  BEQ     LSCOUT          ;YES - GET OUT
026456' 122702 000046           CMPB    #46,R2          ;AMPERSAND ?
026462' 001435                  BEQ     LSCOUT          ;YES - GET OUT
026464' 122702 000040           CMPB    #40,R2          ;SPACE ?
026470' 001752                  BEQ     LSKBL           ;YES - CHECK NEXT CHARACTER
026472' 122702 000011           CMPB    #11,R2          ;TAB ?
026476' 001747                  BEQ     LSKBL           ;YES - CHECK NEXT CHARACTER
026500' 005201          LCTCHR: INC     R1              ;INCREMENT CHAR. COUNT
026502' 112002                  MOVB    (R0)+,R2        ;GET CHAR, ADVANCE POINTER
026504' 122702 000015           CMPB    #15,R2          ;CARRIAGE RETURN ?
026510' 001420                  BEQ     LSBKUP          ;YES - DONE
026512' 122702 000055           CMPB    #55,R2          ;HYPHEN ?
026516' 001415                  BEQ     LSBKUP          ;YES - DONE
026520' 122702 000054           CMPB    #54,R2          ;COMMA ?
026524' 001412                  BEQ     LSBKUP          ;YES - DONE
026526' 122702 000046           CMPB    #46,R2          ;AMPERSAND ?
026532' 001407                  BEQ     LSBKUP          ;YES - DONE
026534' 122702 000040           CMPB    #40,R2          ;SPACE ?
026540' 001415                  BEQ     LSCKGO          ;YES-CHECK FOR GO
026542' 122702 000011           CMPB    #11,R2          ;TAB ?
```

```
 7185  026546' 001401                          BEQ     LSBKUP          ;YES - DONE
 7186  026550' 000753                          BR      LCTCHR          ;NO-GET AND COUNT NEXT CHAR
 7187  026552' 005300                  LSBKUP: DEC     R0              ;BACK UP SCAN POINTER
 7188  026554' 000401                          BR      LSCNEX          ;EXIT
 7189  026556' 005201                  LSCOUT: INC     R1              ;INCREMENT CHARACTER COUNT
 7190  026560' 005300                  LSCNEX: DEC     R0              ;BACK UP SCAN POINTER
 7191  026562' 010067  000036                  MOV     R0,LSCNPT       ;STORE SCAN POINTER
 7192  026566' 010167  000034                  MOV     R1,LSCNCT       ;SAVE SCAN CHAR COUNT
 7193  026572' 000205                          RTS     R5              ;RETURN TO CALLING PROGRAM
 7194  026574' 022701  000002          LSCKGO: CMP     #2,R1           ;CHAR COUNT = 2 ?
 7195  026600' 001364                          BNE     LSBKUP          ;NO-EXIT
 7196  026602' 122760  000107  177775          CMPB    #'G,-3(R0)      ;FIRST CHAR 'G'
 7197  026610' 001360                          BNE     LSBKUP          ;NO-EXIT
 7198  026612' 122760  000117  177776          CMPB    #'O,-2(R0)      ;SECOND CHAR 'O'
 7199  026620' 001354                          BNE     LSBKUP          ;NO-EXIT
 7200  026622' 000726                          BR      LCTCHR          ;YES-GET AND COUNT NEXT CHAR
 7201
 7202
 7203  026624' 000000                  LSCNPT: .WORD   0               ;SCAN POINTER
 7204  026626' 000000                  LSCNCT: .WORD   0               ;SCAN CHARACTER COUNT
 7205  026630' 000000                  LSPTSV: .WORD   0               ;SCAN POINTER SAVE
```

```
                        .SBTTL  TAG LOOKUP SUBROUTINE

                ;*****************************************************************
                ;
                ;       THIS SUBROUTINE SEARCHES THE COMPILER SYMBOL
                ;       TABLES FOR AN ENTRY CORRESPONDING TO A TAG
                ;       ENTERED BY THE USER. IF FOUND, ITS BINARY
                ;       EQUIVALENT IS PASSED BACK IN R2, OTHERWISE
                ;       THE ERROR RETURN IS TAKEN
                ;
                ;       CALLING SEQUENCE-       JSR     R5,TAGLUP
                ;                               .WORD   ERRTN-.
                ;
                ;       ENTRY-  R0  POINTS TO THE FIRST CHAR. OF THE TAG
                ;               R1  CONTAINS CHARACTER COUNT OF THE TAG
                ;               R3  POINTS TO PROGRAM TABLE
                ;
                ;       EXIT-   R2  CONTAINS ABSOLUTE ADDRESS OF TAG
                ;
                ;*****************************************************************

026632' 010167  000524          TAGLUP: MOV     R1,LTAGCT               ;SAVE TAG CHAR. COUNT
026636' 010702                          MOV     PC,R2
026640' 062702  001174                  ADD     #LCSTBL-.,R2            ;POINT R2 AT SYMBOL TABLE
026644' 020267  000506          LTLLP1: CMP     R2,LSYMTE               ;END OF TABLE
026650' 001002                          BNE     LTLEP2
026652' 000167  000274                  JMP     LSYNIT                  ;YES - CHECK DEVICE ROUTINE TBL
026656' 020267  000476          LTLEP2: CMP     R2,LDRSTE               ;END OF DEVICE ROUT TBL ?
026662' 001002                          BNE     LTLEP3
026664' 000167  000312                  JMP     LSYNDT                  ;YES - CHECK INTERF WD TBL
026670' 021227  177777          LTLEP3: CMP     (R2),#177777            ;END OF INTERF WD SYM TBL?
026674' 001442                          BEQ     LSYMNF                  ;YES - SYMBOL NOT FOUND
026676' 121012                          CMPB    (R0),(R2)               ;FIRST CHARS. EQUAL?
026700' 001033                          BNE     LNOTIT                  ;NO - BRANCH
026702' 005301                          DEC     R1                      ;YES - DECREMENT CHAR CNT
026704' 126062  000001  000001          CMPB    1(R0),1(R2)             ;SECOND CHARS. EQUAL?
026712' 001026                          BNE     LNOTIT                  ;NO - BRANCH
026714' 005301                          DEC     R1                      ;YES - DECREMENT CHAR CNT
026716' 001413                          BEQ     LTGLN2                  ;BRANCH IF COUNT=0
026720' 126062  000002  000002          CMPB    2(R0),2(R2)             ;THIRD CHARS. EQUAL?
026726' 001020                          BNE     LNOTIT                  ;NO - BRANCH
026730' 005301                          DEC     R1                      ;YES - DECREMENT CHAR CNT
026732' 001411                          BEQ     LTGLN1                  ;BRANCH IF COUNT=0
026734' 126062  000003  000003          CMPB    3(R0),3(R2)             ;FOURTH CHARS. EQUAL?
026742' 001012                          BNE     LNOTIT                  ;NO - BRANCH
026744' 000422                          BR      LTTBIN                  ;WE FOUND IT
026746' 126227  000002  000040  LTGLN2: CMPB    2(R2),#40               ;THIRD CHAR. A BLANK?
026754' 001005                          BNE     LNOTIT                  ;NO - BRANCH
026756' 126227  000003  000040  LTGLN1: CMPB    3(R2),#40               ;FOURTH CHAR A BLANK?
026764' 001001                          BNE     LNOTIT                  ;NO - BRANCH
026766' 000411                          BR      LTTBIN                  ;YES - WE FOUND IT
026770' 016701  000366          LNOTIT: MOV     LTAGCT,R1               ;RESTORE CHAR. COUNT
026774' 062702  000006                  ADD     #6,R2                   ;BUMP SYMBOL POINTER
027000' 000721                          BR      LTLLP1                  ;TRY AGAIN
027002' 016701  000354          LSYMNF: MOV     LTAGCT,R1               ;RESTORE R1
```

```
7263  027006' 061505                    ADD     (R5),R5         ;CALCULATE ERROR RETURN
7264  027010' 000205                    RTS     R5              ;TAKE ERROR RETURN
7265  027012' 010701            LTTBIN: MOV     PC,R1           ;POINT R1 AT
7266  027014' 062701  000350            ADD     #LTGADR-.,R1    ;ADDRESS AREA
7267  027020' 020267  000324            CMP     R2,LXRTBS       ;R2 BEFORE INDEX REF START?
7268  027024' 103444                    BLO     LDIR            ;MUST BE DIRECT REFERENCE
7269  027026' 020267  000320            CMP     R2,LIRTBS       ;R2 BEFORE INDIRECT REF START?
7270  027032' 103431                    BLO     LINDEX          ;MUST BE INDEXED REFERENCE
7271  027034' 020267  000314            CMP     R2,LIITBS       ;R2 BEFORE INDX IND START ?
7272  027040' 103416                    BLO     LINDIR          ;MUST BE IND REF
7273  027042' 020267  000310            CMP     R2,LSYMTE       ;R2 AFTER MPG TBL END ?
7274  027046' 101402                    BLOS    LININD
7275  027050' 000167  000142            JMP     LDISP           ;MUST BE DISPLACED REFERENCE
7276  027054' 062702  000004    LININD: ADD     #4,R2           ;BUMP TABLE POINTER
7277  027060' 011202                    MOV     (R2),R2         ;GET DISPLACEMENT
7278  027062' 060302                    ADD     R3,R2           ;ADD TO PROG TBL START
7279  027064' 005011                    CLR     (R1)            ;CLR MSW OF ADDR
7280  027066' 011261  000002            MOV     (R2),2(R1)      ;UPDATE LSW OF ADR
7281              001             .IF DF MM
7282                                    TSTB    NPRFLG          ;TEST NPR FLAG
7283                                    BPL     30$             ;IF RESET, SKIP
7284                                    MOV     R3,-(SP)        ;SET UP COMPARISON ADR
7285                                    ADD     #PRDIOA,(SP)
7286                                    CMP     (SP)+,R2        ;DOING RDIO ADR?
7287                                    BNE     10$             ;Y,N-10$
7288                                    ADD     #PRDIOX-PRDIOA,R2 ;POINT AT RDIO 18 BIT ADR
7289                                    BR      20$             ;CONTINUE PROCESSING
7290                            10$:    ADD     #PWRIOX-PWRIOA,R2 ;POINT AT WRIO 18 BIT ADR
7291                            20$:    MOV     (R2)+,(R1)      ;UPDATE MSW OF ADR
7292                                    MOV     (R2)+,2(R1)     ;UPDATE LSW OF ADR
7293                                    BITB    #10,MASS70      ;TEST MASS BUS/70 FLAG
7294                                    BNE     30$             ;IF SET, LEAVE ABS ADR ALONE
7295                                    BIT     #UNIMAP,CSYSFW  ;TEST UNIBUS MAP FLAG
7296                                    BEQ     30$             ;IF NOT SET LEAVE ABS ADR ALONE
7297                                    MOV     (R2)+,(R1)      ;ELSE GET 18 BIT VADR
7298                                    MOV     (R2),2(R1)
7299              000             .ENDC
7300  027072' 000167  000236    30$:    JMP     LTLEXT          ;EXIT
7301  027076' 062702  000004    LINDIR: ADD     #4,R2           ;BUMP TABLE POINTER
7302  027102' 011202                    MOV     (R2),R2         ;GET FREE ADR PTR
7303  027104' 005011                    CLR     (R1)            ;CLR MSW OF ADR
7304  027106' 010261  000002            MOV     R2,2(R1)        ;UPDATE LSW OF ADR
7305              001             .IF DF MM
7306                                    MOV     (R2)+,2(R1)     ;UPDATE LSW OF ADR
7307                                    TSTB    NPRFLG          ;TEST NPR FLAG
7308                                    BPL     10$             ;IF RESET, SKIP
7309                                    MOV     (R2)+,(R1)      ;UPDATE MSW OF ADR
7310                                    MOV     (R2)+,2(R1)     ;UPDATE LSW OF ADR
7311                                    BITB    #10,MASS70      ;TEST MASS BUS/70 FLAG
7312                                    BNE     10$             ;IF SET, LEAVE ABS ADR
7313                                    BIT     #UNIMAP,CSYSFW  ;TEST UNIBUS MAP FLAG
7314                                    BEQ     10$             ;IF NOT SET, LEAVE ABS ADR
7315                                    MOV     (R2)+,(R1)      ;ELSE GET 18 BIT VADR
7316                                    MOV     (R2),2(R1)
7317              000             .ENDC
7318  027112' 000167  000216    10$:    JMP     LTLEXT          ;EXIT
```

```
7319  027116' 062702  000004          LINDEX: ADD     #4,R2                   ;BUMP TABLE POINTER
7320  027122' 011202                          MOV     (R2),R2                 ;GET DISPLACEMENT
7321                  001               .IF DF MM
7322                                            BIC     #P4CONS,R3              ;CLEAR PAR4 SELECT BITS
7323                  000               .ENDC
7324  027124' 060302                           ADD     R3,R2                   ;ADD TO PROG TBL START
7325  027126' 005011                  XCALC:  CLR     (R1)                    ;CLR MSW OF ADR
7326  027130' 010261  000002                  MOV     R2,2(R1)                ;UPDATE LSW OF ADR
7327                  001               .IF DF MM
7328                                            BIC     #100000,2(R1)           ;RESET PAR4 SELECT BITS
7329                                            TSTB    NPRFLG                  ;TEST NPR FLAG
7330                                            BPL     20$                     ;SKIP IF NOT SET
7331                                            MOV     2(R1),R3                ;DISPLACEMENT TO R3
7332                                            MOV     PC,R2                   ;POINT R2 AT
7333                                            ADD     #PGABAD-.,R2            ;P TBL ABS ADR
7334                                            BIT     #UNIMAP,@#SYSFW         ;UNIBUS MAP?
7335                                            BEQ     10$                     ;NO, GO CALC ABS ADR
7336                                            BITB    #10,NPRFLG              ;MASS BUS DEV?
7337                                            BNE     10$                     ;YES, GO CALC ABSADR
7338                                            ADD     #4,R2                   ;ADJUST R2 FOR PTBL VADR
7339                                    10$:    MOV     (R2)+,(R1)              ;GET APPROPRIATE
7340                                            MOV     (R2),2(R1)              ;P TBL ADR
7341                                            ADD     R3,2(R1)
7342                                            ADC     (R1)
7343                                            BR      20$
7344                  000               .ENDC
7345  027134' 000477                    20$:    BR      LTLEXT                  ;EXIT
7346  027136' 062702  000004            LDIR:   ADD     #4,R2                   ;BUMP TABLE POINTER
7347  027142' 005011                            CLR     (R1)                    ;CLR MSW OF ADR
7348  027144' 011261  000002                    MOV     (R2),2(R1)              ;UPDATE LSW OF ADR
7349                  001               .IF DF MM
7350                                            TSTB    NPRFLG                  ;TEST NPR FLAG
7351                                            BPL     10$                     ;SKIP IF RESET
7352                                            BIT     #UNIMAP,@#SYSFW         ;UNIBUS MAP?
7353                                            BEQ     5$                      ;NO, GO CALC ABS ADR
7354                                            BITB    #10,NPRFLG              ;MASS BUS DEV?
7355                                            BEQ     10$                     ;NO, LEAVE VADR ALONE
7356                                    5$:     SUB     #P6CONS,2(R1)           ;TAKE OUT P6 BITS
7357                                            MOV     PC,R2
7358                                            ADD     #PG6BA-.,R2             ;PICK UP PAGE 6 BASE ADR
7359                                            ADD     R2,2(R1)                ;ADD TO DISPLACEMENT
7360                                            ADC     (R1)                    ;UPDATE MSW WITH CARRY
7361                  000               .ENDC
7362  027150' 000471                    10$:    BR      LTLEXT                  ;EXIT
7363  027152' 010302                    LSYNIT: MOV     R3,R2                   ;POINT R2 AT PROG TBL
7364  027154' 062702  000346                    ADD     #PTLGTH+DEVETP,R2
7365  027160' 061202                            ADD     (R2),R2                 ;ENTRY LENGTH PNTR TO R2
7366  027162' 010267  000172                    MOV     R2,LDRSTE               ;USE AS SYMBOL TABLE END
7367  027166' 010302                            MOV     R3,R2                   ;POINT R2 AT PROG TABLE
7368  027170' 062702  000344                    ADD     #PTLGTH+DEVSTP,R2
7369  027174' 061202                            ADD     (R2),R2                 ;SYMBOL TBL POINTER TO R2
7370  027176' 000167  177454                    JMP     LTLEP2                  ;GO LOOKUP SYMBOL
7371  027202' 010302                    LSYNDT: MOV     R3,R2                   ;POINT R2 AT PROG TBL
7372  027204' 062702  000356                    ADD     #PTLGTH+DVIWSP,R2
7373  027210' 061202                            ADD     (R2),R2                 ;INTERFACE SYM TBL ADR TO R2
7374  027212' 000167  177452                    JMP     LTLEP3                  ;GO LOOK UP SYMBOL
```

```
7375  027216' 010301           LDISP:  MOV     R3,R1                   ;POINT R1 AT PROG TBL
7376  027220' 062701  000356           ADD     #PTLGTH+DVIWSP,R1
7377  027224' 061101                   ADD     (R1),R1                 ;THEN TO START OF I W SYM TBL
7378  027226' 020201                   CMP     R2,R1                   ;R2 > R1?
7379  027230' 103020                   BHIS    LIWS                    ;MUST POINT AT INTERF WORD
7380  027232' 010301                   MOV     R3,R1                   ;POINT R1 AT PROG TBL
7381  027234' 062701  000266           ADD     #PTLGTH+DEVDRA,R1
7382  027240' 062702  000004           ADD     #4,R2                   ;R1 POINTS TO DEV REG BASE ADDR
7383  027244' 011202                   MOV     (R2),R2                 ;R2 HAS DISPLACEMENT
7384  027246' 100023                   BPL     LCOMB                   ;IF + COMBINE WITH BASE
7385                                                   ;*****DQ11 SPECIAL CODE STARTS HERE
7386  027250' 042702  100000           BIC     #100000,R2              ;STRIP OFF MINUS SIGN
7387  027254' 010701                   MOV     PC,R1                   ;POINT R1
7388  027256' 062701  000106           ADD     #LTGADR-.,R1            ;AT ADDRESS AREA
7389  027262' 005011                   CLR     (R1)                    ;CLEAR MSW OF ADR
7390  027264' 010261  000002           MOV     R2,2(R1)                ;UPDATE LSW OF ADR
7391  027270' 000421                   BR      LTLEXT                  ;PASS ALONG TABLE VALUE
7392                                                   ;*****DQ11 SPECIAL CODE ENDS HERE
7393  027272' 062702  000004   LIWS:   ADD     #4,R2                   ;BUMP TABLE POINTER
7394  027276' 011202                   MOV     (R2),R2                 ;GET DISPLACEMENT
7395  027300' 060302                   ADD     R3,R2                   ;ADD TO PROG TBL START
7396  027302' 062702  000242           ADD     #PTLGTH,R2              ;ADD PROG TBL LENGTH
7397  027306' 010701                   MOV     PC,R1                   ;POINT R1
7398  027310' 062701  000054           ADD     #LTGADR-.,R1            ;AT ADDRESS AREA
7399  027314' 000704                   BR      XCALC                   ;GO STORE INTERFACE WORD ADR
7400  027316' 061102           LCOMB:  ADD     (R1),R2                 ;COMBINE BASE + DISP
7401  027320' 010701                   MOV     PC,R1                   ;POINT R1
7402  027322' 062701  000042           ADD     #LTGADR-.,R1            ;AT ADDRESS AREA
7403  027326' 005011                   CLR     (R1)                    ;STORE
7404  027330' 010261  000002           MOV     R2,2(R1)                ;16 BIT VADR
7405              001                  .IF DF MM
7406                                           TSTB    NPRFLG          ;NPR DEVICE?
7407                                           BPL     LTLEXT          ;NO, LEAVE 16 BIT VADR
7408                                           BIT     #BITS22,CSYSFW  ;USING 22 BIT ADR?
7409                                           BNE     10$             ;YES, GO HANDLE IT
7410                                           ADD     #3,(R1)         ;ELSE ADJUST TO 18 BIT
7411                                           BR      LTLEXT
7412                                    10$:   ADD     #77,(R1)        ;ADJUST TO 22 BIT
7413              000                  .ENDC
7414  027334' 010102           LTLEXT: MOV     R1,R2                   ;POINT R2 AT ADR AREA
7415  027336' 016701  000020           MOV     LTAGCT,R1               ;RESTORE R1
7416  027342' 005725                   TST     (R5)+
7417  027344' 000205                   RTS     R5                      ;RETURN TO CALLING PGM
7418
7419  027346' 030034'          LDRTBS: .WORD   LCSTBL                  ;DIRECT REF TABLE START
7420  027350' 030414'          LXRTBS: .WORD   LXTBL                   ;INDEXED REF TABLE START
7421  027352' 030620'          LIRTBS: .WORD   LINDTB                  ;INDIRECT REF TABLE START
7422  027354' 030634'          LIITBS: .WORD   LXINDT
7423  027356' 030650'          LSYMTE: .WORD   LSMEND                  ;SYMBOL TABLE END
7424  027360' 000000           LDRSTE: .WORD   0                       ;DEVICE ROUTINE SYM TBL END
7425  027362' 000000           LTAGCT: .WORD   0                       ;TAG CHARACTER COUNT
7426  027364' 000000  000000   LTGADR: .WORD   0,0                     ;ADDRESS TO BE PASSED BACK
```

```
                                        .SBTTL  FIND LINE ENTRY SUBROUTINE

                                ;*****************************************************************
                                ;
                                ;       THIS SUBROUTINE SEARCHES THE PACKED SOURCE ENTRIES
                                ;       FOR A GIVEN LINE NUMBER.  IF IT IS NOT THERE, IT
                                ;       TAKES THE "NOT FOUND" EXIT.
                                ;
                                ;       CALLING SEQUENCE        JSR     R5,LFINLN
                                ;                               .WORD   NOTFND-.
                                ;
                                ;       ENTRY-  R3 POINTS TO PROGRAM TABLE
                                ;               R2 POINTS TO NEW LINE (2 BYTES BEFORE LINE #)
                                ;
                                ;       EXIT-   R1 POINTS TO THE REQUESTED LINE
                                ;
                                ;*****************************************************************

027370' 016301  000022          LFINLN: MOV     PSRCST(R3),R1           ;GET SOURCE START
027374' 016300  000024          LFLLP1: MOV     POBJST(R3),R0           ;GET OBJECT START
            001                 .IF DF MM
                                        BIC     #P4CONS,R0              ;RESET PAR4 SELECT BITS
            000                 .ENDC
027400' 020100                          CMP     R1,R0                   ;IS THIS THE END?
027402' 001407                          BEQ     LENDFD                  ;YES, EXIT
            001                 .IF DF MM
                                        BIS     #P4CONS,R1              ;SET PAR 4 SELECT BITS
            000                 .ENDC
027404' 026162  000002  000002          CMP     2(R1),2(R2)             ;IS THIS THE LINE?
027412' 001405                          BEQ     LINFND                  ;YES-EXIT
027414' 111100                          MOVB    (R1),R0                 ;GET SOURCE LENGTH
            001                 .IF DF MM
                                        BIC     #P4CONS,R1              ;RESET PAR4 SELECT BITS
            000                 .ENDC
027416' 060001                          ADD     R0,R1                   ;POINT R1 AT NEXT ENTRY
027420' 000765                          BR      LFLLP1                  ;TRY AGAIN
            001                 .IF DF MM
                                .IFT
                                LENDFD: BIS     #P4CONS,R1              ;SET PAR4 SELECT BITS IN LINE ADR
                                        ADD     (R5),R5                 ;CALC NOT FOUND EXIT
                                .IFF
027422' 061505                  LENDFD: ADD     (R5),R5                 ;CALC NOT FOUND EXIT
            000                 .ENDC
027424' 000205                          RTS     R5                      ;AND GO THERE
027426' 005725                  LINFND: TST     (R5)+
027430' 000205                          RTS     R5                      ;RETURN TO CALLING PGM
```

```
                                        .SBTTL  OBJECT SIZER SUBROUTINE

                                ;*****************************************************************
                                ;
                                ;       THIS SUBROUTINE DETERMINES THE AMOUNT
                                ;       OF MEMORY REQUIRED FOR THE MACHINE CODE
                                ;       THAT IS GENERATED BY A GIVEN STATEMENT.
                                ;
                                ;       CALLING SEQUENCE        JSR     R5,LCKOBL
                                ;
                                ;       ENTRY-  R0 POINTS TO PACKED STATEMENT
                                ;
                                ;       EXIT-   R0 CONTAINS LENGTH
                                ;
                                ;*****************************************************************

027432' 010146                  LCKOBL: MOV     R1,-(SP)                ;SAVE
027434' 010246                          MOV     R2,-(SP)                ;R1 AND R2
027436' 010702                          MOV     PC,R2
027440' 062702  001212                  ADD     #LCMCTB-.,R2            ;POINT R2 AT COMPILER TABLE
027444' 105760  000001          LCOLP1: TSTB    1(R0)                   ;IF FUNCT CODE NOT = 0
027450' 001002                          BNE     LCOTG1                  ;CHECK TABLE
027452' 005000                          CLR     R0                      ;OTHERWISE SET LENGTH = 0
027454' 000464                          BR      LCOTG2
027456' 122762  000240  000001  LCOTG1: CMPB    #240,1(R2)              ;F.C. REPRESENTS END?
027464' 001405                          BEQ     LCOEIT                  ;YES - CHECK FURTHER
027466' 126062  000001  000001          CMPB    1(R0),1(R2)             ;COMPARE FUNCTION CODES
027474' 001433                          BEQ     LCOLFD                  ;BRANCH IF FOUND
027476' 000426                          BR      LCOBPT                  ;ELSE BUMP POINTER
027500' 126062  000001  000001  LCOEIT: CMPB    1(R0),1(R2)             ;COMPARE FUNCTION CODES
027506' 001426                          BEQ     LCOLFD                  ;MUST BE END
027510' 016702  171554                  MOV     LPKPTP,R2               ;GET P TBL POINTER
027514' 062702  000354                  ADD     #PTLGTH+DVCTEP,R2       ;POINT R2
027520' 061202                          ADD     (R2),R2                 ;AT COMPILE TBL EXTEN.
027522' 122762  000377  000001  10$:    CMPB    #377,1(R2)              ;END OF TBL
027530' 001410                          BEQ     LCOBUG                  ;YES-MUST BE BUG
027532' 126062  000001  000001          CMPB    1(R0),1(R2)             ;COMPARE FUNCTION CODES
027540' 001411                          BEQ     LCOLFD                  ;BRANCH IF FOUND
027542' 111201                          MOVB    (R2),R1
027544' 006301                          ASL     R1
027546' 060102                          ADD     R1,R2                   ;ELSE BUMP POINTER
027550' 000764                          BR      10$                     ;TRY AGAIN
027552' 000000                  LCOBUG: HALT                            ;YES-HALT (MPG BUG)
027554' 111201                  LCOBPT: MOVB    (R2),R1
027556' 006301                          ASL     R1
027560' 060102                          ADD     R1,R2                   ;BUMP TABLE POINTER
027562' 000730                          BR      LCOLP1                  ;TRY AGAIN
027564' 122760  000241  000001  LCOLFD: CMPB    #241,1(R0)              ;WAS IT A PRINT * ?
027572' 001001                          BNE     LCOCNR                  ;NO-GET LENGTH FROM TABLE
027574' 000404                          BR      LCOGPL                  ;YES-GET PACKED LENGTH
027576' 122760  000261  000001  LCOCNR: CMPB    #261,1(R0)              ;WAS IT A PRINT # ?
027604' 001004                          BNE     LCOGLT                  ;NO-GET LENGTH FROM TABLE
027606' 111000                  LCOGPL: MOVB    (R0),R0                 ;YES-GET PACKED LENGTH
027610' 062700  000002                  ADD     #2,R0                   ;THEN ADD 2
027614' 000404                          BR      LCOTG2                  ;AND EXIT
```

```
 7532  027616' 111200                LCOGLT: MOVB    (R2),R0
 7533  027620' 006300                        ASL     R0              ;CONVERT LENGTH TO BYTES
 7534  027622' 162700  000002                SUB     #2,R0           ;AND ADJUST
 7535  027626' 012602                LCOTG2: MOV     (SP)+,R2        ;RESTORE
 7536  027630' 012601                        MOV     (SP)+,R1        ;R1 AND R2
 7537  027632' 000205                        RTS     R5              ;RETURN TO CALLING PGM
```

```
                                .SBTTL  FIND DEVICE COMMAND ENTRY SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE SEARCHES THE DEVICE ROUTINE
                        ;       ENTRY TABLE FOR A FUNCTION CODE MATCH.
                        ;       IF IT FINDS A MATCH IT RETURNS WITH THE
                        ;       ABSOLUTE ADDRESS OF THE ENTRY POINT
                        ;       IN R2.  IF THERE IS NO MATCH, IT POINTS
                        ;       AN ERROR MESSAGE IN THE FOLLOWING FORMAT:
                        ;       'CANT XXXX ON YYYY', WHERE XXXX IS THE
                        ;       ATTEMPTED COMMAND AND YYYY IS THE DEVICE NAME.
                        ;       IT THEN TAKES THE ERROR RETURN.
                        ;
                        ;       CALLING SEQUENCE-       JSR     R5,FENTRY
                        ;                               .WORD   ERRTN-.
                        ;
                        ;       ENTRY   R0  CONTAINS FUNCTION CODE
                        ;               R2  POINTS AT PROGRAM TABLE
                        ;
                        ;       EXIT-   R2  CONTAINS ABSOLUTE ADDRESS OF ENTRY
                        ;               R1  CONTAINS NPR FLAG BYTE
                        ;               R0  IS DESTROYED
                        ;
                        ;*****************************************************************

027634' 010201          FENTRY: MOV     R2,R1                   ;POINT R1 AT PROG TBL
027636' 062702 000346           ADD     #PTLGTH+DEVETP,R2       ;GET ENTRY TBL POINTER
027642' 061202                  ADD     (R2),R2                 ;POINT R2 AT TABLE
027644' 120027 000167           CMPB    R0,#167                 ;IF NOT RD, WR OR BRK
027650' 101002                  BHI     LENLP1                  ;CONTINUE
027652' 042700 000007           BIC     #7,R0                   ;ELSE RESET BITS 0-2
027656' 122712 000377   LENLP1: CMPB    #377,(R2)               ;END OF TABLE?
027662' 001413                  BEQ     LENCNT                  ;TYPE 'CANT DO IT' MSG
027664' 121200                  CMPB    (R2),R0                 ;FUNCTION CODE IN TBL
027666' 001403                  BEQ     LENFCF                  ;YES - BRANCH
027670' 062702 000004           ADD     #4,R2                   ;NO - BUMP POINTER
027674' 000770                  BR      LENLP1                  ;KEEP LOOKING
027676' 116201 000001   LENFCF: MOVB    1(R2),R1                ;GET NPR FLAG
027702' 005722                  TST     (R2)+                   ;R2 = ENTRY DISPLACEMENT
027704' 061202                  ADD     (R2),R2                 ;CALC ABSOLUTE ADDR
                    001         .IF DF MM
                                SUB     #P4CONS,R2              ;TAKE OUT PAR 4 BITS
                    000         .ENDC
027706' 005725                  TST     (R5)+                   ;BYPASS ERROR RETURN ADR
027710' 000205                  RTS     R5                      ;EXIT
027712' 010702          LENCNT: MOV     PC,R2
027714' 062702 171374           ADD     #LPKTBL+6-.,R2          ;POINT R2 AT PACK TBL
027720' 120012          LENLP2: CMPB    R0,(R2)                 ;FUNCTION CODE MATCH?
027722' 001406                  BEQ     LENCFD                  ;YES - COMMAND FOUND
027724' 122712 000377           CMPB    #377,(R2)               ;END OF TABLE?
027730' 001424                  BEQ     LENEEX                  ;YES - TAKE ERROR EXIT
027732' 062702 000010           ADD     #8.,R2                  ;BUMP R2
027736' 000770                  BR      LENLP2                  ;KEEP LOOKING
027740' 162702 000006   LENCFD: SUB     #6,R2                   ;R2 POINTS TO ASCII CMD
```

```
7595  027744' 012267  000044                    MOV     (R2)+,LENCMD            ;MOVE
7596  027750' 012267  000042                    MOV     (R2)+,LENCMD+2          ;ASCII COMMAND
7597  027754' 012267  000040                    MOV     (R2)+,LENCMD+4          ;TO ERROR MESSAGE
7598  027760' 016101  000032                    MOV     PMDLCD(R1),R1           ;GET MDL CODE IN R1
7599  027764' 004567  163556                    JSR     R5,FDVNAM               ;FIND DEV NAME FOR THIS MDL
7600  027770' 000036                            .WORD   LENDEV-.                ;CODE & PUT IN THE MSG
7601  027772' 004567  156226                    JSR     R5,CTYPE                ;TYPE 'CANT XXXX ON YYYY'
7602  027776' 000402                            .WORD   402
7603  030000' 000006                            .WORD   LENMSG-.
7604  030002' 061505                  LENEEX:   ADD     (R5),R5                 ;CALC ERROR RETURN
7605  030004' 000205                            RTS     R5                      ;AND TAKE IT
7606
7607  030006' 040503  023516  020124  LENMSG:   .ASCII  /CAN'T /
7608  030014' 020040  020040  020040  LENCMD:   .ASCII  /       ON /
      030022' 047440  020116
7609  030026' 020040  020040     000  LENDEV:   .ASCIZ  /    /
7610          030034'                           .EVEN
7611
```

```
                                        .SBTTL  COMMON STORAGE SYMBOL TABLE (DIRECT REFERENCE)


                    001                 .IF NDF MM
030034' 052502  020106          LCSTBL: .ASCII  /BUF /
030040' 041440'                         .WORD   CBUF00
030042' 043102  030060                  .ASCII  /BF00/
030046' 041440'                         .WORD   CBUF00
030050' 043102  030460                  .ASCII  /BF01/
030054' 041460'                         .WORD   CBUF01
030056' 043102  031060                  .ASCII  /BF02/
030062' 041500'                         .WORD   CBUF02
030064' 043102  031460                  .ASCII  /BF03/
030070' 041520'                         .WORD   CBUF03
030072' 043102  032060                  .ASCII  /BF04/
030076' 041540'                         .WORD   CBUF04
030100' 043102  032460                  .ASCII  /BF05/
030104' 041560'                         .WORD   CBUF05
030106' 043102  033060                  .ASCII  /BF06/
030112' 041600'                         .WORD   CBUF06
030114' 043102  033460                  .ASCII  /BF07/
030120' 041620'                         .WORD   CBUF07
030122' 043102  034060                  .ASCII  /BF08/
030126' 041640'                         .WORD   CBUF08
030130' 043102  034460                  .ASCII  /BF09/
030134' 041660'                         .WORD   CBUF09
030136' 043102  030061                  .ASCII  /BF10/
030142' 041700'                         .WORD   CBUF10
030144' 043102  030461                  .ASCII  /BF11/
030150' 041720'                         .WORD   CBUF11
030152' 043102  031061                  .ASCII  /BF12/
030156' 041740'                         .WORD   CBUF12
030160' 043102  031461                  .ASCII  /BF13/
030164' 041760'                         .WORD   CBUF13
030166' 043102  032061                  .ASCII  /BF14/
030172' 042000'                         .WORD   CBUF14
030174' 043102  032461                  .ASCII  /BF15/
030200' 042020'                         .WORD   CBUF15
```

```
 7652  030202' 047503  030115          LCTBST: .ASCII  /COM0/
 7653  030206' 042040'                         .WORD   LCOM0
 7654  030210' 047503  030515                  .ASCII  /COM1/
 7655  030214' 042042'                         .WORD   LCOM1
 7656  030216' 047503  031115                  .ASCII  /COM2/
 7657  030222' 042044'                         .WORD   LCOM2
 7658  030224' 047503  031515                  .ASCII  /COM3/
 7659  030230' 042046'                         .WORD   LCOM3
 7660  030232' 047503  032115                  .ASCII  /COM4/
 7661  030236' 042050'                         .WORD   LCOM4
 7662  030240' 047503  032515                  .ASCII  /COM5/
 7663  030244' 042052'                         .WORD   LCOM5
 7664  030246' 047503  033115                  .ASCII  /COM6/
 7665  030252' 042054'                         .WORD   LCOM6
 7666  030254' 047503  033515                  .ASCII  /COM7/
 7667  030260' 042056'                         .WORD   LCOM7
 7668  030262' 047503  034115                  .ASCII  /COM8/
 7669  030266' 042060'                         .WORD   LCOM8
 7670  030270' 047503  034515                  .ASCII  /COM9/
 7671  030274' 042062'                         .WORD   LCOM9
 7672  030276' 040520  030124                  .ASCII  /PAT0/
 7673  030302' 042064'                         .WORD   LPAT0P
 7674  030304' 040520  030524                  .ASCII  /PAT1/
 7675  030310' 042066'                         .WORD   LPAT1P
 7676  030312' 040520  031124                  .ASCII  /PAT2/
 7677  030316' 042070'                         .WORD   LPAT2P
 7678  030320' 040520  031524                  .ASCII  /PAT3/
 7679  030324' 042072'                         .WORD   LPAT3P
 7680  030326' 040520  032124                  .ASCII  /PAT4/
 7681  030332' 042074'                         .WORD   LPAT4P
 7682  030334' 040520  032524                  .ASCII  /PAT5/
 7683  030340' 042076'                         .WORD   LPAT5P
 7684  030342' 040520  033124                  .ASCII  /PAT6/
 7685  030346' 042100'                         .WORD   LPAT6P
 7686  030350' 040520  033524                  .ASCII  /PAT7/
 7687  030354' 042102'                         .WORD   LPAT7P
 7688  030356' 040520  034124                  .ASCII  /PAT8/
 7689  030362' 042104'                         .WORD   LPAT8P
 7690  030364' 040520  034524                  .ASCII  /PAT9/
 7691  030370' 042106'                         .WORD   LPAT9P
 7692  030372' 040520  040524                  .ASCII  /PATA/
 7693  030376' 042110'                         .WORD   LPATAP
 7694  030400' 040520  041124                  .ASCII  /PATB/
 7695  030404' 042112'                         .WORD   LPATBP
 7696  030406' 051103  043114                  .ASCII  /CRLF/
 7697  030412' 042114'                         .WORD   LCRETN
 7698            000                           .ENDC
 7699            001                           .IF DF MM
 7700                                  LCSTBL: .ASCII  /BUF /
 7701                                          .WORD   CBUF00-PG6BA+P6CONS
 7702                                          .ASCII  /BF00/
 7703                                          .WORD   CBUF00-PG6BA+P6CONS
 7704                                          .ASCII  /BF01/
 7705                                          .WORD   CBUF01-PG6BA+P6CONS
 7706                                          .ASCII  /BF02/
 7707                                          .WORD   CBUF02-PG6BA+P6CONS
```

```
7708                                    .ASCII  /BF03/
7709                                    .WORD   CBUF03-PG6BA+P6CONS
7710                                    .ASCII  /BF04/
7711                                    .WORD   CBUF04-PG6BA+P6CONS
7712                                    .ASCII  /BF05/
7713                                    .WORD   CBUF05-PG6BA+P6CONS
7714                                    .ASCII  /BF06/
7715                                    .WORD   CBUF06-PG6BA+P6CONS
7716                                    .ASCII  /BF07/
7717                                    .WORD   CBUF07-PG6BA+P6CONS
7718                                    .ASCII  /BF08/
7719                                    .WORD   CBUF08-PG6BA+P6CONS
7720                                    .ASCII  /BF09/
7721                                    .WORD   CBUF09-PG6BA+P6CONS
7722                                    .ASCII  /BF10/
7723                                    .WORD   CBUF10-PG6BA+P6CONS
7724                                    .ASCII  /BF11/
7725                                    .WORD   CBUF11-PG6BA+P6CONS
7726                                    .ASCII  /BF12/
7727                                    .WORD   CBUF12-PG6BA+P6CONS
7728                                    .ASCII  /BF13/
7729                                    .WORD   CBUF13-PG6BA+P6CONS
7730                                    .ASCII  /BF14/
7731                                    .WORD   CBUF14-PG6BA+P6CONS
7732                                    .ASCII  /BF15/
7733                                    .WORD   CBUF15-PG6BA+P6CONS
7734                                    .PAGE
7735
7736
7737                            LCTBST: .ASCII  /COM0/
7738                                    .WORD   LCOM0-PG6BA+P6CONS
7739                                    .ASCII  /COM1/
7740                                    .WORD   LCOM1-PG6BA+P6CONS
7741                                    .ASCII  /COM2/
7742                                    .WORD   LCOM2-PG6BA+P6CONS
7743                                    .ASCII  /COM3/
7744                                    .WORD   LCOM3-PG6BA+P6CONS
7745                                    .ASCII  /COM4/
7746                                    .WORD   LCOM4-PG6BA+P6CONS
7747                                    .ASCII  /COM5/
7748                                    .WORD   LCOM5-PG6BA+P6CONS
7749                                    .ASCII  /COM6/
7750                                    .WORD   LCOM6-PG6BA+P6CONS
7751                                    .ASCII  /COM7/
7752                                    .WORD   LCOM7-PG6BA+P6CONS
7753                                    .ASCII  /COM8/
7754                                    .WORD   LCOM8-PG6BA+P6CONS
7755                                    .ASCII  /COM9/
7756                                    .WORD   LCOM9-PG6BA+P6CONS
7757                                    .ASCII  /PAT0/
7758                                    .WORD   LPAT0P-PG6BA+P6CONS
7759                                    .ASCII  /PAT1/
7760                                    .WORD   LPAT1P-PG6BA+P6CONS
7761                                    .ASCII  /PAT2/
7762                                    .WORD   LPAT2P-PG6BA+P6CONS
7763                                    .ASCII  /PAT3/
```

```
7764                                    .WORD   LPAT3P-PG6BA+P6CONS
7765                                    .ASCII  /PAT4/
7766                                    .WORD   LPAT4P-PG6BA+P6CONS
7767                                    .ASCII  /PAT5/
7768                                    .WORD   LPAT5P-PG6BA+P6CONS
7769                                    .ASCII  /PAT6/
7770                                    .WORD   LPAT6P-PG6BA+P6CONS
7771                                    .ASCII  /PAT7/
7772                                    .WORD   LPAT7P-PG6BA+P6CONS
7773                                    .ASCII  /PAT8/
7774                                    .WORD   LPAT8P-PG6BA+P6CONS
7775                                    .ASCII  /PAT9/
7776                                    .WORD   LPAT9P-PG6BA+P6CONS
7777                                    .ASCII  /PATA/
7778                                    .WORD   LPATAP-PG6BA+P6CONS
7779                                    .ASCII  /PATB/
7780                                    .WORD   LPATBP-PG6BA+P6CONS
7781                                    .ASCII  /CRLF/
7782                                    .WORD   LCRETN-PG6BA+P6CONS
7783                    000             .ENDC
```

```
                                        .SBTTL  TEMPORARY STORAGE SYMBOL TABLE (INDEX REFERENCED)

030414' 046524  030060          LXTBL:  .ASCII  /TM00/
030420' 000056                          .WORD   PTEM0
030422' 046524  030460                  .ASCII  /TM01/
030426' 000060                          .WORD   PTEM1
030430' 046524  031060                  .ASCII  /TM02/
030434' 000062                          .WORD   PTEM2
030436' 046524  031460                  .ASCII  /TM03/
030442' 000064                          .WORD   PTEM3
030444' 046524  032060                  .ASCII  /TM04/
030450' 000066                          .WORD   PTEM4
030452' 046524  032460                  .ASCII  /TM05/
030456' 000070                          .WORD   PTEM5
030460' 046524  033060                  .ASCII  /TM06/
030464' 000072                          .WORD   PTEM6
030466' 046524  033460                  .ASCII  /TM07/
030472' 000074                          .WORD   PTEM7
030474' 046524  034060                  .ASCII  /TM08/
030500' 000076                          .WORD   PTEM8
030502' 046524  034460                  .ASCII  /TM09/
030506' 000100                          .WORD   PTEM9
030510' 046524  030061                  .ASCII  /TM10/
030514' 000102                          .WORD   PTEM10
030516' 046524  030461                  .ASCII  /TM11/
030522' 000104                          .WORD   PTEM11
030524' 046524  031061                  .ASCII  /TM12/
030530' 000106                          .WORD   PTEM12
030532' 046524  031461                  .ASCII  /TM13/
030536' 000110                          .WORD   PTEM13
030540' 046524  032061                  .ASCII  /TM14/
030544' 000112                          .WORD   PTEM14
030546' 046524  032461                  .ASCII  /TM15/
030552' 000114                          .WORD   PTEM15
030554' 050117  053523                  .ASCII  /OPSW/
030560' 000002                          .WORD   POPSW
030562' 044523  042532                  .ASCII  /SIZE/
030566' 000262                          .WORD   PTLGTH+DEVIW7
030570' 051105  020122                  .ASCII  /ERR /
030574' 000264                          .WORD   PTLGTH+DEVIW8
030576' 041116  020122                  .ASCII  /NBR /
030602' 000116                          .WORD   PNBR
030604' 051123  020103                  .ASCII  /SRC /
030610' 000120                          .WORD   PSRC
030612' 051504  020124                  .ASCII  /DST /
030616' 000122                          .WORD   PDST
```

```
                                        .SBTTL  INDIRECT REFERENCE SYMBOL TABLE

030620' 044515  046104          LINDTB: .ASCII  /MIDL/
            001                         .IF DF MM
                                        .IFF
030624' 177777                  LMIDL:  .WORD   177777
                                        .IFT
                                LMIDL:  .WORD   MIDLAD
            000                         .ENDC
030626' 051106  042505                  .ASCII  /FREE/
            001                         .IF DF MM
                                        .IFF
030632' 177777                  LFREE:  .WORD   177777
                                        .IFT
                                LFREE:  .WORD   FREEAD
            000                         .ENDC
030634' 042122  047511          LXINDT: .ASCII  /RDIO/
030640' 000016                          .WORD   PRDIOA
030642' 051127  047511                  .ASCII  /WRIO/
030646' 000020                          .WORD   PWRIOA
030650' 000000                  LSMEND: .WORD   0
            001                         .IF DF MM


                                FREEAD: .WORD   P3CONS                  ;16 BIT VIRTUAL ADR FOR FREE
                                        .WORD   XXXX                    ;18/22 BIT ABSOLUTE ADDRESS
                                        .WORD   XXXX                    ;FOR FREE
                                FREEVA: .WORD   XXXX                    ;18 BIT VIRTUAL ADDRESS
                                        .WORD   XXXX                    ;FOR FREE

                                MIDLAD: .WORD   XXXX                    ;16 BIT VIRTUAL ADR FOR MIDL
                                MIDLAA: .WORD   XXXX                    ;18/22 BIT ABSOLUTE ADDRESS
                                        .WORD   XXXX                    ;FOR MIDL
                                MIDLVA: .WORD   XXXX                    ;18 BIT VIRTUAL ADDRESS
                                        .WORD   XXXX                    ;FOR MIDL
        000                             .ENDC


        001                             .IF DF MM
                                        .IFF
                                        .TITLE  MAINDEC-11-DTMGA-C   MPG COMPILER
                                        .IFT
                                        .TITLE  MAINDEC-11-DTMMA-B   MPG COMPILER
        000                             .ENDC
```

```
                                .SBTTL  COMPILER MACHINE CODE TABLE

                        ;*****************************************************************
                        ;
                        ;       COMPILER UTILIZES TABLE AS FOLLOWS-
                        ;
                        ;       ENTRY   ACTION
                        ;
                        ;      >15      STORES TABLE ENTRY AS OBJECT CODE
                        ;       15      STORES A VALUE OF 1 AS OBJECT CODE
                        ;       14      CALCULATES DST ADDRESS FOR THIS PROGRAM
                        ;       13      CALCULATES SRC ADDRESS FOR THIS PROGRAM
                        ;       12      CALCULATES NBR ADDRESS FOR THIS PROGRAM
                        ;       11      STORES ALL ZEROES AS OBJECT CODE
                        ;       10      FINDS AND STORES DEVICE ROUTINE ENTRY ADDRESS
                        ;        9      STORES A VALUE OF 5 AS OBJECT CODE
                        ;        8      MODIFYS PREVIOUS CODE IF V NOT A TAG
                        ;        7      CALCULATES WRIO ADDRESS FOR THIS PROGRAM
                        ;        6      CALCULATES RDIO ADDRESS FOR THIS PROGRAM
                        ;        5      STORES A VALUE OF 2 AS OBJECT CODE
                        ;        4      INTERPRETS PACKED CODE AS BIT NUMBERS
                        ;        3      MODIFYS NEXT TABLE ENTRY IF V NOT A TAG
                        ;        2      INTERPRETS VARIABLE THEN SWAPS WITH PRECEEDING WORD
                        ;        1      INTERPRETS PACKED CODE AS BIT NUMBER THEN SWAPS
                        ;        0      INTERPRETS PACKED CODE AS VARIABLE-STORES BINARY EQUIV.
                        ;
                        ;*****************************************************************

030652' 004 242         LCMCTB: .BYTE   4,242                   ;LOAD V WITH V
030654' 012737 000000 000002    .WORD   012737,0,2
030662' 003 011                 .BYTE   3,011                   ;INCR V
030664' 005237 000000           .WORD   005237,0
030670' 003 021                 .BYTE   3,021                   ;DECR V
030672' 005337 000000           .WORD   005337,0
030676' 004 013                 .BYTE   4,013                   ;INCR V BY V
030700' 062737 000000 000002    .WORD   062737,0,2
030706' 004 023                 .BYTE   4,023                   ;DECR V BY V
030710' 162737 000000 000002    .WORD   162737,0,2
030716' 012 030                 .BYTE   12,030                  ;MOVE
030720' 013700 000014 013701    .WORD   13700,12.,13701,13.
030726' 000015
030730' 013702 000016 112122    .WORD   13702,14.,112122
030736' 005300 001375           .WORD   5300,1375
030742' 004 033                 .BYTE   4,033                   ;MOVE V TO V
030744' 013737 000000 000000    .WORD   013737,0,0
030752' 012 035                 .BYTE   12,035                  ;MOVE V AT V TO V
030754' 012700 000000 012701    .WORD   012700,0,012701,0
030762' 000000
030764' 012702 000000 112122    .WORD   012702,0,112122
030772' 005300 001375           .WORD   005300,001375
030776' 004 043                 .BYTE   4,043                   ;ADD V TO V
031000' 063737 000000 000000    .WORD   063737,0,0
031006' 004 053                 .BYTE   4,053                   ;SUB V FROM V
031010' 163737 000000 000000    .WORD   163737,0,0
031016' 006 045                 .BYTE   6,045                   ;ADD V AT V TO V
```

```
7933  031020' 004537                          .WORD   004537
7934  031022' 035722' 000000  000000  LRT1:   .WORD   LADDM,0,0,0
      031030' 000000
7935  031032'    006     055                  .BYTE   6,055                   ;SUB V AT V FROM V
7936  031034' 004537                          .WORD   004537
7937  031036' 036020' 000000  000000  LRT2:   .WORD   LSUBM,0,0,0
      031044' 000000
7938  031046'    004     247                  .BYTE   4,247                   ;SET V BIT N
7939  031050' 052737  000000  000001          .WORD   052737,0,1
7940  031056'    004     250                  .BYTE   4,250                   ;CLEAR V BIT N
7941  031060' 042737  000000  000001          .WORD   042737,0,1
7942  031066'    011     100                  .BYTE   9.,100                  ;IF V BIT N SET GO TO LN
7943  031070' 013700  000000  005100          .WORD   013700,0,005100,032700,4
      031076' 032700  000004
7944  031102' 001002  000137  000000          .WORD   001002,000137,0
7945  031110'    007     101                  .BYTE   7,101                   ;IF V BIT N CLEAR GO TO LN
7946  031112' 032737  000000  000001          .WORD   032737,0,1
7947  031120' 001002  000137  000000          .WORD   001002,000137,0
7948  031126'    007     102                  .BYTE   7,102                   ;IF V = V GO TO LN
7949  031130' 023737  000000  000010          .WORD   023737,0,8.
7950  031136' 001002  000137  000000          .WORD   001002,000137,0
7951  031144'    007     103                  .BYTE   7,103                   ;IF V > V GO TO LN
7952  031146' 023737  000000  000010          .WORD   023737,0,8.
7953  031154' 101402  000137  000000          .WORD   101402,000137,0
7954  031162'    007     104                  .BYTE   7,104                   ;IF V < V GO TO LN
7955  031164' 023737  000000  000010          .WORD   023737,0,8.
7956  031172' 103002  000137  000000          .WORD   103002,000137,0
7957  031200'    007     105                  .BYTE   7,105                   ;IF V => V GO TO LN
7958  031202' 023737  000000  000010          .WORD   023737,0,8.
7959  031210' 103402  000137  000000          .WORD   103402,000137,0
7960  031216'    007     106                  .BYTE   7,106                   ;IF V <= V GO TO LN
7961  031220' 023737  000000  000010          .WORD   023737,0,8.
7962  031226' 101002  000137  000000          .WORD   101002,000137,0
7963  031234'    007     107                  .BYTE   7,107                   ;IF V <> V GO TO LN
7964  031236' 023737  000000  000010          .WORD   023737,0,8.
7965  031244' 001402  000137  000000          .WORD   001402,000137,0
7966  031252'    016     112                  .BYTE   16,112                  ;IF V AT V = V GO TO LN
7967  031254' 012700  000000  012701          .WORD   012700,0,012701,0
      031262' 000000
7968  031264' 012702  000003  122122          .WORD   012702,3,122122,001005
      031272' 001005
7969  031274' 000302                          .WORD   000302
7970  031276' 005300  001373  000137          .WORD   005300,001373,000137,0
      031304' 000000
7971  031306'    017     117                  .BYTE   17,117                  ;IF V AT V <> V GO TO LN
7972  031310' 012700  000000  012701          .WORD   012700,0,012701,0
      031316' 000000
7973  031320' 012702  000003  122122          .WORD   012702,3,122122,001004
      031326' 001004
7974  031330' 000302                          .WORD   000302
7975  031332' 005300  001373  000402          .WORD   005300,001373,000402,000137,0
      031340' 000137  000000
7976  031344'    003     244                  .BYTE   3,244                   ;GO TO LN
7977  031346' 000137  000000                  .WORD   000137,0
7978  031352'    003     245                  .BYTE   3,245                   ;LINK LN
7979  031354' 004737  000000                  .WORD   004737,0
```

```
 7980  031360'     002     246                        .BYTE   2,246           ;RETURN
 7981  031362'  000207                                .WORD   000207
 7982  031364'     003     260                        .BYTE   3,260           ;NEGATE
 7983  031366'  005437  000000                        .WORD   005437,0
 7984  031372'     005     261                        .BYTE   5,261           ;PRINT # TEXT
 7985  031374'  004537                                .WORD   4537
 7986  031376'  040022' 031402' 000004  LRT21:  .WORD   LPRNG,.+2,4
 7987  031404'     005     241                        .BYTE   5,241           ;PRINT * TEXT
 7988  031406'  004537                                .WORD   004537
 7989  031410'  040022' 031414' 000004  LRT3:   .WORD   LPRNG,.+2,4
 7990  031416'     005     154                        .BYTE   5,154           ;PRINT V IN ASCII
 7991  031420'  004537                                .WORD   004537
 7992  031422'  040022' 000000  000005  LRT4:   .WORD   LPRNG,0,5
 7993  031430'     005     155                        .BYTE   5,155           ;PRINT V IN OCTAL
 7994  031432'  004537                                .WORD   004537
 7995  031434'  040126' 000000  000005  LRT5:   .WORD   LPRNO,0,5
 7996  031442'     005     156                        .BYTE   5,156           ;PRINT V IN DECIMAL
 7997  031444'  004537                                .WORD   004537
 7998  031446'  040114' 000000  000005  LRT6:   .WORD   LPRND,0,5
 7999  031454'     005     157                        .BYTE   5,157           ;PRINT V IN BINARY
 8000  031456'  004537                                .WORD   004537
 8001  031460'  040402' 000000  000005  LRT7:   .WORD   LPRNB,0,5
 8002  031466'     005     150                        .BYTE   5,150           ;PRINT V AT V IN ASCII
 8003  031470'  004537                                .WORD   004537
 8004  031472'  040022' 000000  000002  LRT8:   .WORD   LPRNG,0,2
 8005  031500'     005     151                        .BYTE   5,151           ;PRINT V AT V IN OCTAL
 8006  031502'  004537                                .WORD   004537
 8007  031504'  040126' 000000  000002  LRT9:   .WORD   LPRNO,0,2
 8008  031512'     005     152                        .BYTE   5,152           ;PRINT V AT V IN DECIMAL
 8009  031514'  004537                                .WORD   004537
 8010  031516'  040114' 000000  000002  LRT10:  .WORD   LPRND,0,2
 8011  031524'     005     153                        .BYTE   5,153           ;PRINT V AT V IN BINARY
 8012  031526'  004537                                .WORD   004537
 8013  031530'  040402' 000000  000002  LRT11:  .WORD   LPRNB,0,2
 8014  031536'     005     170                        .BYTE   5,170           ;FILL V WITH RANDOM
 8015  031540'  004537                                .WORD   004537
 8016  031542'  036216' 000000  000005  LRT12:  .WORD   LFILR,0,5
 8017  031550'     005     172                        .BYTE   5,172           ;FILL V WITH ASCII
 8018  031552'  004537                                .WORD   004537
 8019  031554'  036274' 000000  000005  LRT13:  .WORD   LFILG,0,5
 8020  031562'     006     174                        .BYTE   6,174           ;FILL V WITH V
 8021  031564'  004537                                .WORD   004537
 8022  031566'  036152' 000000  000005  LRT14:  .WORD   LFILV,0,5,0
       031574'  000000
 8023  031576'     005     171                        .BYTE   5,171           ;FILL V AT V WITH RANDOM
 8024  031600'  004537                                .WORD   004537
 8025  031602'  036216' 000000  000002  LRT15:  .WORD   LFILR,0,2
 8026  031610'     005     173                        .BYTE   5,173           ;FILL V AT V WITH ASCII
 8027  031612'  004537                                .WORD   004537
 8028  031614'  036274' 000000  000002  LRT16:  .WORD   LFILG,0,2
 8029  031622'     006     175                        .BYTE   6,175           ;FILL V AT V WITH V
 8030  031624'  004537                                .WORD   004537
 8031  031626'  036152' 000000  000002  LRT17:  .WORD   LFILV,0,2,0
       031634'  000000
 8032  031636'     006     176                        .BYTE   6,176           ;FILL V WITH PATN
 8033  031640'  004537                                .WORD   004537
```

```
8034  031642' 036326' 000000  000005  LRT22:  .WORD   LFILP,0,5,0
      031650' 000000
8035  031652'    006     177                  .BYTE   6,177                   ;FILL V AT V WITH PATN
8036  031654' 004537                          .WORD   004537
8037  031656' 036326' 000000  000002  LRT23:  .WORD   LFILP,0,2,0
      031664' 000000
8038  031666'    005     001                  .BYTE   5,001                   ;ROTATE V
8039  031670' 004537                          .WORD   004537
8040  031672' 036732' 000007  000000  LRT18:  .WORD   LROTR,7,0
8041  031700'    005     003                  .BYTE   5,003                   ;ROTATE V AT V
8042  031702' 004537                          .WORD   004537
8043  031704' 036732' 000000  000002  LRT19:  .WORD   LROTR,0,2
8044  031712'    004     243                  .BYTE   4,243                   ;DELAY V
8045  031714' 004537                          .WORD   004537
8046  031716' 036764' 000000          LRT20:  .WORD   LDLAY,0
8047  031722'    007     120                  .BYTE   7,120                   ;READ
8048  031724' 004537                          .WORD   004537
8049  031726' 000012  000400  000006          .WORD   10.,00400,6,6,11.
      031734' 000006  000013
8050  031740'    007     121                  .BYTE   7,121                   ;READ V
8051  031742' 004537                          .WORD   004537
8052  031744' 000012  000000  000006          .WORD   10.,0,6,6,11.
      031752' 000006  000013
8053  031756'    007     123                  .BYTE   7,123                   ;READ V INTO V
8054  031760' 004537                          .WORD   004537
8055  031762' 000012  000000  000002          .WORD   10.,0,2,2,11.
      031770' 000002  000013
8056  031774'    007     125                  .BYTE   7,125                   ;READ V INTO V FROM N
8057  031776' 004537                          .WORD   004537
8058  032000' 000012  000000  000002          .WORD   10.,0,2,2,4
      032006' 000002  000004
8059  032012'    007     130                  .BYTE   7,130                   ;WRITE
8060  032014' 004537                          .WORD   004537
8061  032016' 000012  000400  000007          .WORD   10.,00400,7,7,11.
      032024' 000007  000013
8062  032030'    007     131                  .BYTE   7,131                   ;WRITE V
8063  032032' 004537                          .WORD   004537
8064  032034' 000012  000000  000007          .WORD   10.,0,7,7,11.
      032042' 000007  000013
8065  032046'    007     133                  .BYTE   7,133                   ;WRITE V FROM V
8066  032050' 004537                          .WORD   004537
8067  032052' 000012  000000  000002          .WORD   10.,0,2,2,11.
      032060' 000002  000013
8068  032064'    007     135                  .BYTE   7,135                   ;WRITE V FROM V TO N
8069  032066' 004537                          .WORD   004537
8070  032070' 000012  000000  000002          .WORD   10.,0,2,2,4
      032076' 000002  000004
8071  032102'    006     143                  .BYTE   6,143                   ;VERIFY V WITH V
8072  032104' 004537                          .WORD   004537
8073  032106' 037004' 000005  000000  LRT27:  .WORD   LVERFY,5,0,0
      032114' 000000
8074  032116'    006     145                  .BYTE   6,145                   ;VERIFY V AT V WITH V
8075  032120' 004537                          .WORD   004537
8076  032122' 037004' 000000  000000  LRT28:  .WORD   LVERFY,0,0,0
      032130' 000000
8077  032132'    007     161                  .BYTE   7,161                   ;BREAK V
```

```
 8078  032134' 004537                          .WORD   004537
 8079  032136' 000012  000000  000007          .WORD   10.,0,7,7,11.
       032144' 000007  000013
 8080  032150'    007     163                  .BYTE   7,163                   ;BREAK V ON V
 8081  032152' 004537  000012                  .WORD   4537,10.
 8082  032156' 000000  000007  000007          .WORD   0,7,7,4
       032164' 000004
 8083                                  .IF NDF MM
 8084  032166'    004     251                  .BYTE   4,251                   ;VECTOR V TO LN
 8085  032170' 012737  000000  000002          .WORD   012737,0,2
 8086                                  .ENDC
 8087  032176'    003     252                  .BYTE   3,252                   ;LETGO
 8088  032200' 004537                          .WORD   004537
 8089  032202' 035216'                 LRT29:  .WORD   CUPCR
 8090  032204'    007     253                  .BYTE   7,253                   ;ENTRY
 8091  032206' 010546  010446  010346          .WORD   10546,10446,10346
 8092  032214' 010246  010146  010046          .WORD   10246,10146,10046
 8093  032222'    010     254                  .BYTE   8.,254                  ;EXIT
 8094  032224' 012600  012601  012602          .WORD   12600,12601,12602
 8095  032232' 012603  012604  012605          .WORD   12603,12604,12605,5
       032240' 000005
 8096                                  .IF NDF MM
 8097  032242'    002     255                  .BYTE   2,255                   ;RESET
 8098  032244' 000011                          .WORD   9.
 8099                                  .ENDC
 8100  032246'    002     262                  .BYTE   2,262                   ;PAUSE
 8101  032250' 000017                          .WORD   15.
 8102  032252'    002     263                  .BYTE   2,263                   ;WORD V
 8103  032254' 000000                          .WORD   0
 8104  032256'    003     240                  .BYTE   3,240                   ;END (MUST BE LAST IN THIS TABLE)
 8105  032260' 004537                          .WORD   004537
 8106  032262' 035016'                 LRT30:  .WORD   CUPGEX
```

```
                                .SBTTL  MPG COMPILER

                        ;*******************************************************************
                        ;
                        ;       THIS ROUTINE CONVERTS THE INTERNAL
                        ;       PACKED CODE INTO MACHINE LANGUAGE
                        ;       AND STORES IT IN MEMORY.
                        ;
                        ;       CALLING SEQUENCE - ENTERED FROM UPLP
                        ;
                        ;       ENTRY - R3 POINTS TO PROGRAM TABLE
                        ;
                        ;       EXIT - RETURNS TO CALLING PROGRAM (CONTROL ROUTINE)
                        ;
                        ;*******************************************************************

032264' 010446          LCPILE: MOV     R4,-(SP)                ;SAVE R4
032266' 010367  166776          MOV     R3,LPKPTP               ;SAVE P TBL POINTER
032272' 042713  000100          BIC     #OCPRES,(R3)            ;RESET OBJ CODE PRES FLG
032276' 016304  000022          MOV     PSRCST(R3),R4           ;POINT R4 AT PACKED SOURCE
            001                 .IF DF MM
                                BIS     #P4CONS,R4              ;SELECT PAR4
            000                 .ENDC
032302' 016302  000024          MOV     POBJST(R3),R2           ;POINT R2 AT OBJECT ADDR
032306' 116401  000001  LCPLP1: MOVB    1(R4),R1                ;GET FUNCTION CODE
032312' 010264  000004          MOV     R2,4(R4)                ;STORE PHYSICAL ADDRESS
032316' 010400                  MOV     R4,R0                   ;SET PKED STMNT PNTR IN R0
032320' 004567  175106          JSR     R5,LCKOBL               ;GO FIND OBJECT LENGTH
032324' 060002                  ADD     R0,R2                   ;ADD TO OBJECT ADDRESS
032326' 122701  000240          CMPB    #240,R1                 ;WAS IT 'END'
032332' 001403                  BEQ     LCPEFP                  ;YES-END OF FIRST PASS
032334' 111400                  MOVB    (R4),R0                 ;GET PACKED LENGTH
032336' 060004                  ADD     R0,R4                   ;BUMP POINTER TO NEXT ENTRY
032340' 000762                  BR      LCPLP1                  ;DO IT AGAIN
            001                 .IF DF MM
                                .IFT
                        LCPEFP: BIS     #P4CONS,R2              ;SET PAR4 SELECT BITS
                                MOV     R2,LCPOEN               ;STORE AS OBJ END
                                .IFF
032342' 010267  001564  LCPEFP: MOV     R2,LCPOEN               ;STORE AS OBJ END
            000                 .ENDC
032346' 160302                  SUB     R3,R2                   ;CALC PROG LENGTH
032350' 010263  000026          MOV     R2,PLNGTH(R3)           ;AND PUT IN TABLE
            001                 .IF NDF MM
032354' 016767  147066  176250  MOV     CFMST,LFREE             ;UPDATE SYMBOL TABLE
032362' 016704  147062          MOV     CFMEND,R4               ;FREE MEM END ADDR TO R4
032366' 005724                  TST     (R4)+                   ;ADD 2 TO IT
032370' 166704  147052          SUB     CFMST,R4                ;SUBTRACT FREE MEM START
032374' 000241                  CLC                             ;DIVIDE BY 2
032376' 006004                  ROR     R4
032400' 042704  000001          BIC     #1,R4                   ;MAKE IT AN EVEN ADDR
032404' 066704  147036          ADD     CFMST,R4                ;CALC MIDDLE OF FREE
032410' 010467  176210          MOV     R4,LMIDL                ;STORE IN SYMBOL TABLE
            000                 .ENDC
032414' 010367  001514          MOV     R3,LCPPTS               ;SAVE R3
```

```
8164  032420' 016304  000022                    MOV     PSRCST(R3),R4           ;POINT R4 AT PACKED SOURCE
8165                  001                       .IF DF MM
8166                                            BIS     #P4CONS,R4              ;SELECT PAR4
8167                  000                       .ENDC
8168  032424' 016367  000024  001506            MOV     POBJST(R3),LCPOBA       ;SET OBJ ADDR=OBJ START
8169                  001                       .IF DF MM
8170                                            BIS     #P4CONS,LCPOBA          ;SELECT PAR4
8171                  000                       .ENDC
8172  032432' 010703                  LCPGNE:  MOV     PC,R3
8173  032434' 062703  176216                    ADD     #LCMCTB-.,R3            ;POINT R3 AT COMPILE TBL
8174  032440' 116401  000001          LCPGFC:  MOVB    1(R4),R1                ;GET FUNCTION CODE
8175  032444' 122701  000241                    CMPB    #241,R1                 ;WAS IT 'PRINT IMM'
8176  032450' 001003                            BNE     LCPCNR                  ;NO-CONTINUE
8177  032452' 105067  001453                    CLRB    LCPNRF                  ;CLEAR NO-RETURN FLAG
8178  032456' 000405                            BR      LCPPIB                  ;YES-HANDLE SPECIAL CASE
8179  032460' 122701  000261          LCPCNR:  CMPB    #261,R1                 ;WAS IT PRINT WITH NO RETN
8180  032464' 001004                            BNE     LCPFTE                  ;NO-CONTINUE
8181  032466' 105267  001437                    INCB    LCPNRF                  ;SET NO-RETURN FLAG
8182  032472' 000167  001336          LCPPIB:  JMP     LCPPRI                  ;GO HANDLE SPECIAL CASE
8183  032476' 122763  000240  000001  LCPFTE:  CMPB    #240,1(R3)              ;F.C. REPRESENTS END?
8184  032504' 001405                            BEQ     LCPEIT                  ;YES - CHECK FURTHER
8185  032506' 126463  000001  000001            CMPB    1(R4),1(R3)             ;COMPARE FUNCTION CODES
8186  032514' 001433                            BEQ     LCPFFC                  ;BRANCH IF FOUND
8187  032516' 000426                            BR      LCPBTP                  ;ELSE BUMP POINTER
8188  032520' 126463  000001  000001  LCPEIT:  CMPB    1(R4),1(R3)             ;COMPARE FUNCTION CODES
8189  032526' 001426                            BEQ     LCPFFC                  ;MUST BE END
8190  032530' 016703  166534                    MOV     LPKPTP,R3               ;GET P TBL POINTER
8191  032534' 062703  000354                    ADD     #PTLGTH+DVCTEP,R3       ;POINT R2
8192  032540' 061303                            ADD     (R3),R3                 ;AT COMPILE TBL EXTEN.
8193  032542' 122763  000377  000001  10$:     CMPB    #377,1(R3)              ;END OF TBL?
8194  032550' 001410                            BEQ     LCPBUG                  ;YES - MUST BE BUG
8195  032552' 126463  000001  000001            CMPB    1(R4),1(R3)             ;COMPARE FUNCTION CODES
8196  032560' 001411                            BEQ     LCPFFC                  ;BRANCH IF FOUND
8197  032562' 111300                            MOVB    (R3),R0
8198  032564' 006300                            ASL     R0
8199  032566' 060003                            ADD     R0,R3                   ;ELSE BUMP POINTER
8200  032570' 000764                            BR      10$                     ;TRY AGAIN
8201  032572' 000000                  LCPBUG:  HALT                            ;YES-HALT (MPG BUG)
8202  032574' 111300                  LCPBTP:  MOVB    (R3),R0                 ;GET ENTRY LENGTH
8203  032576' 006300                           ASL     R0                      ;CONVERT TO BYTES
8204  032600' 060003                           ADD     R0,R3                   ;BUMP POINTER TO NEXT ENTRY
8205  032602' 000735                           BR      LCPFTE                  ;TRY AGAIN
8206  032604' 011467  001332          LCPFFC:  MOV     (R4),LCPTM1
8207  032610' 012714  177777                   MOV     #177777,(R4)
8208  032614' 010400                           MOV     R4,R0
8209  032616' 005200                           INC     R0
8210  032620' 010701                           MOV     PC,R1
8211  032622' 062701  001363                   ADD     #LCPERL-.,R1
8212  032626' 004567  006172                   JSR     R5,DECASC               ;PUT LINE # IN ERR MSG
8213  032632' 016714  001304                   MOV     LCPTM1,(R4)
8214  032636' 062704  000006                   ADD     #6,R4                   ;ADJUST SOURCE POINTER
8215  032642' 112367  001270                   MOVB    (R3)+,LCPTEC            ;GET ENTRY LENGTH
8216  032646' 112367  001272                   MOVB    (R3)+,LCPFCS            ;SAVE FUNCTION CODE
8217  032652' 005367  001260                   DEC     LCPTEC                  ;AND COUNT
8218  032656' 105067  001246                   CLRB    LCPSWP                  ;CLEAR SWAP FLAG
8219  032662' 105067  001236                   CLRB    LDVRIF                  ;CLEAR DEVR INST FLAG
```

```
8220  032666' 011302                  LCPGTE: MOV     (R3),R2                 ;GET TABLE ENTRY
8221  032670' 105067  001231                  CLRB    L2CODF                  ;RESET 2 CODE FLAG
8222  032674' 020227  000017                  CMP     R2,#15.
8223  032700' 101064                          BHI     JCPTSF                  ;IF >15 GO TEST SWAP FLAG
8224
8225                                          ;PROCESS COMPILER CODES 0 THRU 15.
8226
8227  032702' 005702                  LCPTG0: TST     R2
8228  032704' 001450                          BEQ     JCPGNV                  ;IF 0 GET NEXT VARIABLE
8229
8230  032706' 005302                  LCPTG1: DEC     R2
8231  032710' 001005                          BNE     LCPTG2
8232  032712' 112767  000001  001210         MOVB    #1,LCPSWP               ;SET SWAP FLAG
8233  032720' 012402                          MOV     (R4)+,R2                ;PACKED ENTRY TO R2
8234  032722' 000453                          BR      JCPTSF
8235
8236  032724' 005302                  LCPTG2: DEC     R2
8237  032726' 001006                          BNE     LCPTG3
8238  032730' 112767  000001  001172         MOVB    #1,LCPSWP               ;SET SWAP FLAG
8239  032736' 105267  001163                  INCB    L2CODF                  ;SET THE 2 CODE FLAG
8240  032742' 000431                          BR      JCPGNV
8241
8242  032744' 005302                  LCPTG3: DEC     R2
8243  032746' 001031                          BNE     LCPTG4
8244  032750' 122714  000040                  CMPB    #40,(R4)                ;IF FIRST BYTE BLANK
8245  032754' 001416                          BEQ     LCPNT1                  ;IT IS NOT A TAG
8246  032756' 116400  000001                  MOVB    1(R4),R0                ;GET SECOND BYTE
8247  032762' 042700  000017                  BIC     #17,R0                  ;OF PACKED VARIABLE
8248  032766' 122700  000360                  CMPB    #360,R0                 ;IF LEFT 4 BITS ARE ONES
8249  032772' 001407                          BEQ     LCPNT1                  ;IT IS NOT A TAG
8250  032774' 012763  122122  000002         MOV     #122122,2(R3)           ;MODIFY NEXT TABLE ENTRY
8251  033002' 012763  000240  000006         MOV     #240,6(R3)              ;SET UP A NOP INST
8252  033010' 000406                          BR      JCPGNV                  ;GO GET NEXT VARIABLE
8253  033012' 012763  122102  000002  LCPNT1: MOV     #122102,2(R3)           ;MODIFY NEXT TABLE ENTRY
8254  033020' 012763  000302  000006         MOV     #302,6(R3)              ;SET UP "SWAB R2" INST
8255  033026' 000167  000334          JCPGNV: JMP     LCPGNV                  ;GO GET NEXT VARIABLE
8256
8257  033032' 005302                  LCPTG4: DEC     R2
8258  033034' 001002                          BNE     LCPTG5
8259  033036' 012402                          MOV     (R4)+,R2                ;PACKED ENTRY TO R2
8260  033040' 000404                          BR      JCPTSF
8261
8262  033042' 005302                  LCPTG5: DEC     R2
8263  033044' 001004                          BNE     LCPTG6
8264  033046' 012702  000002                  MOV     #2,R2                   ;SET R2=2
8265  033052' 000167  000650          JCPTSF: JMP     LCPTSF                  ;GO TEST SWAP FLAG
8266
8267  033056' 005302                  LCPTG6: DEC     R2
8268  033060' 001013                          BNE     LCPTG7
8269  033062' 016702  001046                  MOV     LCPPTS,R2
8270  033066' 062702  000016                  ADD     #PRDIOA,R2              ;ADDRESS OF READ AREA TO R2
8271  033072' 010701                          MOV     PC,R1                   ;POINT R1
8272  033074' 062701  174270                  ADD     #LTGADR-.,R1            ;AT ADDRESS AREA
8273  033100' 005011                          CLR     (R1)                    ;CLR MSW OF ADR
8274  033102' 011261  000002                  MOV     (R2),2(R1)              ;UPDATE LSW OF ADR
8275              001          .IF DF MM
```

```
8276
8277  033106' 000420                    .IFF
                                        BR      LCP67E                  ;GO SET UP CORRECT REG
8278                                    .IFT
8279                                    TSTB    NPRFLG                  ;TEST IF NPR DEVICE
8280                                    BPL     LCP67E                  ;IF NOT, SKIP
8281                                    ADD     #PRDIOX-PRDIOA,R2       ;POINT AT 18 BIT ADR
8282                                    BR      LCP67C                  ;GO GET ADR & TEST FLAGS
8283            000                     .ENDC
8284
8285  033110' 005302            LCPTG7: DEC     R2
8286  033112' 001022                    BNE     LCPTG8
8287  033114' 016702 001014             MOV     LCPPTS,R2
8288  033120' 062702 000020             ADD     #PWRIOA,R2              ;ADDRESS OF WRITE AREA TO R2
8289  033124' 126727 001014 000001      CMPB    LCPFCS,#001             ;THIS THE 'ROTATE V' INST?
8290  033132' 001410                    BEQ     SU4ROT                  ;N,Y-SU4ROT
8291  033134' 010701                    MOV     PC,R1                   ;POINT R1
8292  033136' 062701 174226             ADD     #LTGADR-.,R1            ;AT ADDRESS AREA
8293  033142' 005011                    CLR     (R1)                    ;CLR MSW OF ADR
8294  033144' 011261 000002             MOV     (R2),2(R1)              ;UPDATE LSW OF ADR
8295            001                     .IF DF MM
8296                                    TSTB    NPRFLG                  ;TEST IF NPR DEVICE
8297                                    BPL     LCP67E                  ;IF NOT, SKIP
8298                                    ADD     #PWRIOX-PWRIOA,R2       ;POINT AT 18 BIT ADR
8299                            LCP67C: MOV     (R2)+,(R1)
8300                                    MOV     (R2)+,2(R1)             ;UPDATE ADDRESS AREA
8301                                    BITB    #10,MASS70              ;TEST MASS BUS/70 FLAG
8302                                    BNE     LCP67E                  ;IF SET, LEAVE ABS ADR ALONE
8303                                    BIT     #UNIMAP,CSYSFW          ;TEST UNIBUS MAP FLAG
8304                                    BEQ     LCP67E                  ;IF NOT SET, LEAVE ABS ADR ALONE
8305                                    MOV     (R2)+,(R1)              ;ELSE GET 18 BIT VADR
8306                                    MOV     (R2),2(R1)
8307            000                     .ENDC
8308  033150' 010102            LCP67E: MOV     R1,R2
8309  033152' 000543                    BR      ADR18B                  ;GO STORE IN OBJ CODE
8310  033154' 011202            SU4ROT: MOV     (R2),R2                 ;GET WRIO'S 16 BIT ADR
8311  033156' 000502                    BR      BCPTSF                  ;GO TEST SWAP FLAG
8312
8313  033160' 005302            LCPTG8: DEC     R2
8314  033162' 001022                    BNE     LCPTG9
8315  033164' 122714 000040             CMPB    #40,(R4)                ;IF FIRST BYTE BLANK
8316  033170' 001413                    BEQ     LCPNT2                  ;IT IS NOT A TAG
8317  033172' 116400 000001             MOVB    1(R4),R0                ;GET SECOND BYTE
8318  033176' 042700 000017             BIC     #17,R0                  ;OF PACKED VARIABLE
8319  033202' 122700 000360             CMPB    #360,R0                 ;IF LEFT 4 BITS SET
8320  033206' 001404                    BEQ     LCPNT2                  ;IT IS NOT A TAG
8321  033210' 012761 023737 177774      MOV     #023737,-4(R1)          ;MODIFY PREVIOUS CODE
8322  033216' 000463                    BR      LCPGNV                  ;GO GET NEXT VARIABLE
8323  033220' 012761 023727 177774 LCPNT2: MOV  #023727,-4(R1)          ;MODIFY PREVIOUS CODE
8324  033226' 000457                    BR      LCPGNV                  ;GO GET NEXT VARIABLE
8325
8326  033230' 005302            LCPTG9: DEC     R2
8327  033232' 001003                    BNE     LCPT10
8328  033234' 012702 000005             MOV     #5,R2
8329  033240' 000451                    BR      BCPTSF
8330
8331  033242' 005302            LCPT10: DEC     R2
```

```
8332  033244' 001014                  BNE     LCPT11
8333  033246' 116700  000672          MOVB    LCPFCS,R0               ;PICK UP  FUNCTION CODE
8334  033252' 016702  000656          MOV     LCPPTS,R2               ;PICK UP PT POINTER
8335  033256' 004567  174352          JSR     R5,FENTRY               ;FIND DEVICE ROUT ENTRY
8336  033262' 000416                  .WORD   LCPABT-.                ;GO HERE IF NOT THERE
8337  033264' 105267  000634          INCB    LDVRIF                  ;SET THE DVR INST FLAG
8338  033270' 110167  000633          MOVB    R1,NPRFLG               ;UPDATE NPR FLAG
8339                  001       .IF DF MM
8340                                  BIT     #CPU70,CSYSFW           ;SYSTEM = 11/70?
8341                                  BEQ     10$                     ;NO, SKIP
8342                                  MOVB    NPRFLG,R1               ;ELSE SET MASS70 FLAG
8343                                  BIC     #177767,R1              ;EQUAL TO
8344                                  MOVB    R1,MASS70               ;MASS BUS DEV FLAG
8345                  000       .ENDC
8346  033274' 000433          10$:    BR      BCPTSF
8347
8348  033276' 005302          LCPT11: DEC     R2
8349  033300' 001002                  BNE     LCPT12
8350  033302' 005002                  CLR     R2                      ;SET R2 = 0
8351  033304' 000427                  BR      BCPTSF
8352
8353  033306' 005302          LCPT12: DEC     R2
8354  033310' 001005                  BNE     LCPT13
8355  033312' 016702  000616          MOV     LCPPTS,R2
8356                  001       .IF DF MM
8357                                  BIC     #P4CONS,R2              ;SELECT PAR0
8358                  000       .ENDC
8359  033316' 062702  000116          ADD     #PNBR,R2                ;ADDR OF NBR TO R2
8360  033322' 000420                  BR      BCPTSF
8361
8362  033324' 005302          LCPT13: DEC     R2
8363  033326' 001005                  BNE     LCPT14
8364  033330' 016702  000600          MOV     LCPPTS,R2
8365                  001       .IF DF MM
8366                                  BIC     #P4CONS,R2              ;SELECT PAR0
8367                  000       .ENDC
8368  033334' 062702  000120          ADD     #PSRC,R2                ;ADDR OF SRC TO R2
8369  033340' 000411                  BR      BCPTSF
8370
8371  033342' 005302          LCPT14: DEC     R2
8372  033344' 001005                  BNE     LCPT15
8373  033346' 016702  000562          MOV     LCPPTS,R2
8374                  001       .IF DF MM
8375                                  BIC     #P4CONS,R2              ;SELECT PAR0
8376                  000       .ENDC
8377  033352' 062702  000122          ADD     #PDST,R2                ;ADDR OF DST TO R2
8378  033356' 000402                  BR      BCPTSF                  ;GO TEST SWAP FLAG
8379
8380  033360' 012702  000001  LCPT15: MOV     #1,R2
8381  033364' 000560          BCPTSF: BR      LCPTSF                  ;GO TEST SWAP FLAG
8382
8383                          ;GET NEXT VARIABLE FROM PACKED SOURCE
8384
8385  033366' 122714  000040  LCPGNV: CMPB    #40,(R4)                ;IF FIRST BYTE IS A BLANK
8386  033372' 001452                  BEQ     LCPOCT                  ;VARIABLE IS AN OCTAL #
8387  033374' 116400  000001          MOVB    1(R4),R0                ;GET SECOND BYTE
```

```
8388  033400' 042700 000017          BIC     #17,R0                  ;OF PACKED VARIABLE
8389  033404' 122700 000360          CMPB    #360,R0                 ;IF LEFT FOUR BITS ARE ONES
8390  033410' 001451                 BEQ     LCPNAL                  ;IT IS NOT A TAG
8391  033412' 010400                 MOV     R4,R0
8392  033414' 012701 000004          MOV     #4,R1
8393  033420' 010367 000516          MOV     R3,LCPTM1
8394  033424' 016703 000504          MOV     LCPPTS,R3
8395  033430' 004567 173176          JSR     R5,TAGLUP               ;LOOK UP TAG
8396  033434' 000234                 .WORD   LCPERR-.
8397  033436' 016703 000500          MOV     LCPTM1,R3
8398  033442' 062704 000004          ADD     #4,R4                   ;ADJUST SOURCE POINTER
8399  033446' 105767 000455          TSTB    NPRFLG                  ;TEST IF 2 WD ADR NEEDED
8400  033452' 001003                 BNE     ADR18B                  ;BRANCH IF YES
8401  033454' 016202 000002          MOV     2(R2),R2                ;PUT TAG VALUE IN R2
8402  033460' 000522                 BR      LCPTSF                  ;THEN GO TEST SWAP FLAG
8403
8404                                  ;TWO WORD ADDRESS PROCESSING
8405
8406  033462' 005367 000450  ADR18B: DEC     LCPTEC                  ;DECREMENT TBL END CTR
8407  033466' 005723                 TST     (R3)+                   ;BUMP COMPILER TBL PTR
8408  033470' 105067 000434          CLRB    LCPSWP
8409  033474' 105067 000427          CLRB    NPRFLG
8410  033500' 016700 000434          MOV     LCPOBA,R0               ;POINT R0 AT OBJ CODE
8411  033504' 014046                 MOV     -(R0),-(SP)             ;SAVE OLD CONTENTS
8412  033506' 012220                 MOV     (R2)+,(R0)+             ;STORE MSW OF ADR
8413  033510' 011220                 MOV     (R2),(R0)+              ;STORE LSW OF ADR
8414  033512' 012620                 MOV     (SP)+,(R0)+             ;PUT BYTE COUNT HERE
8415  033514' 010001                 MOV     R0,R1
8416  033516' 000517                 BR      LCPTEN                  ;CONT LIKE ORIG MPG
8417
8418                                  ;OCTAL OR DECIMAL # OR A LINE #
8419
8420  033520' 005204         LCPOCT: INC     R4
9421  033522' 010400                 MOV     R4,R0
8422  033524' 004567 006402          JSR     R5,OCTBIN               ;CONVERT OCTAL TO BINARY
8423  033530' 000140                 .WORD   LCPERR-.
8424  033532' 000413                 BR      LCK2WD                  ;GO CK IF 2 WORDS NEEDED
8425  033534' 121427 000114  LCPNAL: CMPB    (R4),#'L                ;IS FIRST CHAR AN 'L'
8426  033540' 001434                 BEQ     LCPFLN                  ;YES-GO FIND LINE NBR
8427  033542' 121427 000104          CMPB    (R4),#'D                ;IS FIRST CHAR A 'D'
8428  033546' 001050                 BNE     LCPERR                  ;NO-SOMETHINGS WRONG
8429  033550' 005204                 INC     R4
8430  033552' 010400                 MOV     R4,R0
8431  033554' 004567 006336          JSR     R5,DECBIN               ;CONVERT DECIMAL TO BINARY
8432  033560' 000110                 .WORD   LCPERR-.
8433  033562' 062704 000003  LCK2WD: ADD     #3,R4
8434  033566' 105767 000332          TSTB    LDVRIF                  ;THIS A DEV ROUT INST?
8435  033572' 001415                 BEQ     10$                     ;Y,N-10$
8436  033574' 105767 000327          TSTB    NPRFLG                  ;2 WORD ADR NEEDED?
8437  033600' 001412                 BEQ     10$                     ;Y,N-10$
8438  033602' 105767 000317          TSTB    L2CODF                  ;PROCESSING A 2 CODE?
8439  033606' 001407                 BEQ     10$                     ;Y,N-10$
8440  033610' 010702                 MOV     PC,R2                   ;SET UP ADR OF ADR STORAGE
8441  033612' 062702 173552          ADD     #LTGADR-.,R2
8442  033616' 005012                 CLR     (R2)                    ;CLEAR MSW OF ADR
8443  033620' 010062 000002          MOV     R0,2(R2)                ;STORE LSW OF ADR
```

```
8444  033624' 000716                  BR      ADR18B          ;GO STORE 2 WORDS
8445  033626' 010002          10$:    MOV     R0,R2
8446  033630' 000436                  BR      LCPTSF          ;GO TEST SWAP FLAG
8447  033632' 010402          LCPFLN: MOV     R4,R2
8448  033634' 010367  000302          MOV     R3,LCPTM1
8449  033640' 016703  000270          MOV     LCPPTS,R3
8450  033644' 004567  173520          JSR     R5,LFINLN       ;LOOK UP LINE NUMBER
8451  033650' 000020                  .WORD   LCPERR-.
8452  033652' 016703  000264          MOV     LCPTM1,R3
8453  033656' 062704  000004          ADD     #4,R4
8454  033662' 016102  000004          MOV     4(R1),R2
8455  033666' 000417                  BR      LCPTSF          ;GO TEST SWAP FLAG
8456
8457  033670' 004567  152330  LCPERR: JSR     R5,CTYPE        ;TYPE NON EXIST LINE MSG
8458  033674' 000402                  .WORD   402
8459  033676' 000250                  .WORD   LCPEMS-.
8460  033700' 016701  000230  LCPABT: MOV     LCPPTS,R1
8461  033704' 016100  000024          MOV     POBJST(R1),R0
8462              001         .IF DF MM
8463                          BIS     #P4CONS,R0      ;SELECT PAR4
8464              000         .ENDC
8465  033710' 160100                  SUB     R1,R0           ;SUBTRACT OBJECT LENGTH
8466  033712' 010061  000026          MOV     R0,PLNGTH(R1)   ;FROM PROGRAM LENGTH
8467  033716' 016100  000024          MOV     POBJST(R1),R0   ;PASS FREE MEM START
8468              001         .IF DF MM
8469                          BIS     #P4CONS,R0      ;SELECT PAR4
8470              000         .ENDC
8471  033722' 012604                  MOV     (SP)+,R4        ;RESTORE R4
8472  033724' 000205                  RTS     R5              ;RETURN TO CALLING PGM
8473
8474                          ;TEST THE SWAP FLAG
8475
8476  033726' 016701  000206  LCPTSF: MOV     LCPOBA,R1       ;RECOVER OBJECT ADDRESS
8477  033732' 105767  000172          TSTB    LCPSWP          ;TEST THE SWAP FLAG
8478  033736' 001406                  BEQ     LCPNSW          ;BRANCH IF NO SWAP
8479  033740' 014100                  MOV     -(R1),R0        ;GET PREVIOUS WORD CONTENTS
8480  033742' 010221                  MOV     R2,(R1)+        ;STORE NEW CODE AT PREV. ADR
8481  033744' 010021                  MOV     R0,(R1)+        ;STORE OLD CODE A NEW ADR
8482  033746' 105067  000156          CLRB    LCPSWP          ;RESET SWAP FLAG
8483  033752' 000401                  BR      LCPTEN          ;GO TEST FOR TABLE END
8484  033754' 010221          LCPNSW: MOV     R2,(R1)+        ;STORE NEW CODE AT NEW ADR
8485  033756' 010167  000156  LCPTEN: MOV     R1,LCPOBA       ;PUT BACK NEXT OBJ ADR
8486  033762' 005367  000150          DEC     LCPTEC          ;DECREMENT TABLE ENTRY CNT
8487  033766' 001404                  BEQ     LCPTOE          ;IF ZERO-TEST OBJ END
8488  033770' 062703  000002          ADD     #2,R3           ;BUMP TABLE POINTER
8489  033774' 000167  176666          JMP     LCPGTE          ;GO CHECK NEXT ENTRY
8490  034000' 026767  000134  000124  LCPTOE: CMP     LCPOBA,LCPOEN   ;COMPILE COMPLETE
8491  034006' 001402                  BEQ     LCPOUT          ;YES-GET OUT
8492  034010' 000167  176416          JMP     LCPGNE          ;NO-CHECK NEXT PACKED ENTRY
8493  034014' 016701  000114  LCPOUT: MOV     LCPPTS,R1
8494  034020' 052711  000100          BIS     #OCPRES,(R1)    ;SET OBJ CODE PRESENT FLAG
8495  034024' 016700  000102          MOV     LCPOEN,R0       ;PASS FREE MEM START
8496  034030' 012604                  MOV     (SP)+,R4        ;RESTORE R4
8497  034032' 000205                  RTS     R5              ;RETURN TO CALLING PGM
8498
8499                          ;SPECIAL HANDLING FOR PRINT # AND PRINT *
```

```
 8500
 8501  034034' 016701 000100          LCPPRI: MOV     LCPOBA,R1               ;GET CURRENT OBJ ADDR
 8502  034040' 012721 004537                  MOV     #4537,(R1)+             ;STORE JSR
 8503  034044' 016721 175340                  MOV     LRT3,(R1)+              ;TO PRINT ASCII PGM
 8504  034050' 010100                         MOV     R1,R0
 8505  034052' 062700 000004                  ADD     #4,R0
 8506             001                  .IF DF MM
 8507                                          BIC     #P4CONS,R0              ;SELECT PAR0
 8508             000                  .ENDC
 8509  034056' 010021                         MOV     R0,(R1)+                ;STORE INLINE ADDRESS
 8510  034060' 111400                         MOVB    (R4),R0
 8511  034062' 162700 000006                  SUB     #6,R0                   ;CALCULATE BYTE COUNT
 8512  034066' 010011                         MOV     R0,(R1)                 ;STORE IT
 8513  034070' 005321                         DEC     (R1)+                   ;DECR BY 1
 8514  034072' 062704 000006                  ADD     #6,R4
 8515  034076' 105767 000027                  TSTB    LCPNRF                  ;TEST NO-RETURN FLAG
 8516  034102' 001402                         BEQ     LCPMTX                  ;CONTINUE IF NOT SET
 8517  034104' 005461 177776                  NEG     -2(R1)                  ;OTHERWISE COMPLIMENT BC
 8518  034110' 112421                 LCPMTX: MOVB    (R4)+,(R1)+             ;MOVE TEXT INLINE
 8519  034112' 005300                         DEC     R0
 8520  034114' 001375                         BNE     LCPMTX
 8521  034116' 010167 000016                  MOV     R1,LCPOBA               ;SAVE NEXT OBJ ADDR
 8522  034122' 000726                         BR      LCPTOE                  ;GO TEST FOR OBJ END
 8523
 8524  034124'    000                 LDVRIF: .BYTE   0                       ;DEVICE ROUTINE INST FLAG
 8525  034125'    000                 L2CODF: .BYTE   0                       ;COMPILER 2 CODE FLAG
 8526  034126'    000                 MASS70: .BYTE   0                       ;MASS BUS DEV ON 11/70 FLAG
 8527  034127'    000                 NPRFLG: .BYTE   0                       ;NPR FLAG ADDRESS
 8528  034130'    000                 LCPSWP: .BYTE   0                       ;SWAP FLAG
 8529  034131'    000                 LCPNRF: .BYTE   0                       ;PRINT NO RETURN FLAG
 8530  034132' 000000                 LCPOEN: .WORD   0                       ;OBJECT END ADDRESS
 8531  034134' 000000                 LCPPTS: .WORD   0                       ;PROG TABLE POINTER SAVE
 8532  034136' 000000                 LCPTEC: .WORD   0                       ;TABLE ENTRY COUNT
 8533  034140' 000000                 LCPOBA: .WORD   0                       ;OBJECT CODE ADDRESS
 8534  034142' 000000                 LCPTM1: .WORD   0
 8535  034144' 000000                 LCPFCS: .WORD   0                       ;FUNCTION CODE SAVE
 8536
 8537  034146' 047516 026516 054105   LCPEMS: .ASCII  /NON-EXISTENT LINE REF ON LINE  /
       034154' 051511 042524 052116
       034162' 046040 047111 020105
       034170' 042522 020106 047117
       034176' 046040 047111 020105
       034204'    040
 8538  034205'    040    040    040   LCPERL: .BYTE   40,40,40,40,0,0
       034210'    040    000    000
 8539             034214'                     .EVEN
 8540                001                      .IF DF MM
 8541
 8542
 8543                                 ALPT1=  .-CLOCZ
 8544                                 ALFCT1= ALPT1&000077/2
 8545                002                      .IF NE ALFCT1
 8546                                 ALQUN1= 31.-ALFCT1
 8547                                         .BLKW   ALQUN1                  ;ALIGN TO A 32 WORD BNDRY
 8548                                         .WORD   0
 8549                001                      .ENDC
```

```
8550            000                     .ENDC
8551
8552            001                     .IF DF MM
8553                                    .IFF
8554                                    .TITLE  MAINDEC-11-DTMGA-C   MPG / USER PROGRAM SHARED CODE
8555                                    .IFT
8556                                    .TITLE  MAINDEC-11-DTMMA-B   MPG / USER PROGRAM SHARED CODE
8557            000                     .ENDC
```

```
                                        .SBTTL  SHARED CODE CONSTANTS


000020                          MAXPRG= 16.                             ;MAXIMUM # OF USER PROGRAMS

   001                                  .IF DF MM
                                        .IFT
                                PG6BA:                                  ;SHARED CODE PAGE BASE ADDRESS


                                USTKWK: .BLKW   MAXPRG+1*4              ;USER PROGRAM'S STACK
                                USRSTK:


                                CSYSFW: .WORD   1                       ;MPG'S SYSTEM FLAGWORD
                                        .IFF
034214' 000000                  CSYSFW: .WORD   0                       ;MPG'S SYSTEM FLAGWORD
      000                               .ENDC


000001                                  MMVER=  1                       ;MPG MEMORY MANAGEMENT VERSION
000002                                  USMTPS= 2                       ;CPU REQUIRES MTPS/MFPS INST'S
000004                                  LSI11=  4                       ;CPU IS AN LSI-11
000010                                  CPU70=  10                      ;CPU = 11/70 (MASSBUS USES ABS ADRS)
   001                                  .IF DF MM
                                        CPU40=  20                      ;CPU = 11/40 (NO MMMR1 OR MMMR3)
                                        UNIMAP= 40                      ;USE THE UNIBUS MAP
                                        BITS22= 100                     ;USE 22 BIT ADDRESSING
      000                               .ENDC
000200                                  CACHE=  200                     ;CACHE MEMORY PRESENT IN CPU
001000                                  CAIDLE= 1000                    ;ALL USER PROGRAMS IDLE FLAG
002000                                  CUSPGR= 2000                    ;USER PROG RUNNING FLAG
004000                                  CHPINT= 4000                    ;'HOLD PRINTER'S INT ENB SET' FLG
010000                                  CNTRLO= 10000                   ;SUPPRESS TYPING - CNTRL/O FLAG
020000                                  CTUDED= 20000                   ;PRINT DEVICE DEDICATED FLAG
040000                                  CRUDED= 40000                   ;REQUEST USER DEDICATED RESET
100000                                  CTRQBP= 100000                  ;CONSOLE REQUEST BEING PROCESSED


034216'                         CRTBL:  .REPT   MAXPRG                  ;*;CONTROL ROUTINE TABLE
                                        .WORD   0                       ;*;ONE WORD PER ENTRY -
                                        .ENDR                           ;*;EACH ENTRY CONTAINS 16 BIT ADR OF
                                                                        ;*;PROG TBL FOR RESIDENT USER PROG
034256' 177777                          .WORD   177777                  ;TABLE TERMINATOR
   001i                                 .IF DF MM


                                CRTEXT= MAXPRG+1*2                      ;TABLE EXTENSION ACCESS VALUE


                                CTBLEX: .REPT   MAXPRG                  ;CONTROL ROUTINE TABLE EXTENSION
                                        .WORD   0                       ;BIT  15 = 1 = FIXED ADR PROG
                                        .ENDR                           ;BITS  8 - 0 = PROG AREA LENGTH
                                        FIXADR= 100000                  ;               (# OF 32 WD BLKS)


                                                                        ;VALUES FOR USER'S PAR'S 3 THRU 7
                                FMPAR3: .WORD   XXXX                    ;FREE MEM PAR'S
                                FMPAR4: .WORD   XXXX                    ;
```

```
 8615                                   FMPAR5: .WORD   XXXX             ;
 8616                                   SPPAR6: .WORD   XXXX             ;SHARED PAGE PAR
 8617                                   IOPAR7: .WORD   007600           ;I/O PAGE PAR
 8618
 8619                                           .WORD   0,0,0            ;FILL WORDS
 8620
 8621                                                                    ;VALUES FOR USER'S PDR'S 3 THRU 7
 8622                                   FMPDR3: .WORD   XXXX             ;FREE MEM PDR'S
 8623                                   FMPDR4: .WORD   XXXX             ;
 8624                                   FMPDR5: .WORD   XXXX             ;
 8625                                   SPPDR6: .WORD   XXXX             ;SHARED PAGE PDR
 8626                                   IOPDR7: .WORD   077406           ;I/O PAGE PDR
 8627                       000                 .ENDC
 8628                                           .EVEN
 8629  034260' 025040                          .ASCII  / */
 8630  034262' 000144                   LINBUF: .BLKB   100.             ;PROGRAM STATEMENT BUFFER
 8631  034426' 005015                          .ASCII  <15><12>
 8632  034430' 005015                   LCRLF:  .ASCII  <15><12>         ;CARRIAGE RETN, LINE FEED
 8633  034432' 000000                           .WORD   0
```

```
                                .SBTTL  USER PROGRAM DISPATCHER

                        ;*****************************************************************************
                        ;
                        ;  THE PROGRAM DISPATCHER ROUTINE WILL BE ENTERED FROM THE
                        ;  COMMAND DECODE ROUTINE WHEN A NULL KEYBOARD ENTRY OR A RUN CMND
                        ;  IS ENTERED. IT WILL SCAN THE PROGRAM SLOTS AND LOOK FOR
                        ;  A USER PROGRAM THAT IS AVAILABLE FOR EXECUTION. IF FOUND,
                        ;  CONTROL WILL BE TRANSFERRED TO THAT USER PROGRAM'S OBJECT
                        ;  CODE. IF A PROGRAM IS NOT AVAILABLE, SCANNING OF THE PROGRAM
                        ;  SLOTS WILL CONTINUE AS LONG AS A PROGRAM IS WAITING FOR
                        ;  I/O TERMINATION. IF NONE ARE WAITING FOR TERMINATION OR
                        ;  AVAILABLE FOR EXECUTION, RETURN WILL BE MADE TO THE
                        ;  COMMAND DECODE ROUTINE WHICH REQUESTS A COMMAND ENTRY.
                        ;
                        ;*****************************************************************************

034434' 010705          CPGDSP: MOV     PC,R5           ;SET UP ADR OF SYSTEM FLGWD
034436' 062705 177556           ADD     #C$YSFW-.,R5
034442' 052715 001000           BIS     #CAIDLE,(R5)    ;SET 'ALL IDLE' FLAG
034446' 010704                  MOV     PC,R4           ;SET UP START ADR OF CNTRL ROUT TBL
034450' 062704 177546           ADD     #CRTBL-.,R4
034454' 005367 000006           DEC     CTOCK           ;DECR T/O CHECK VALUE
034460' 100003                  BPL     CPGTTE          ;IS IT EXHAUSTED? (Y,N-CPGTTE)
034462' 012727                  MOV     (PC)+,(PC)+     ;RESTORE ITS VALUE
034464' 000000          CTOADJ: .WORD   0
034466' 000000          CTOCK:  .WORD   0
034470' 010427          CPGTTE: MOV     R4,(PC)+        ;SAVE CURRENT CNTR ROUT TBL ADR
034472' 000000          CPGCPE: .WORD   0               ;CURRENT CNTRL ROUT TBL ENTRY ADR
034474' 005714                  TST     (R4)            ;THIS PROG SLOT IN USE?
034476' 001442                  BEQ     CPGNXT          ;Y,N-CPGNXT
                                .IF DF MM
                                .IFF
034500' 011403                  MOV     (R4),R3         ;GET PROG TBL ADR
                                .IFT
                                MOV     (R4),@#KPAR4    ;MAP KPAR 4 TO THIS PROG
                                MOV     #P4CONS,R3      ;SET UP ITS VIRTUAL ADR
                                MOV     SPPAR6,@#KPAR6  ;MAP KPAR 6 TO SHARED CODE
                                .ENDC
034502' 005713                  TST     (R3)            ;IS PROG'S ACTIVE FLAG SET?
034504' 100037                  BPL     CPGNXT          ;Y,N-CPGNXT
034506' 005027                  CLR     (PC)+           ;RESET T/O ERR ROUT ADR
034510' 000000          TOEADR: .WORD   0
034512' 032713 000004           BIT     #ERSTOP,(R3)    ;ERROR STOPPED THIS PROG?
034516' 001032                  BNE     CPGNXT          ;N,Y-CPGNXT
034520' 032713 000002           BIT     #URSTOP,(R3)    ;USER STOP THIS PROG?
034524' 001403                  BEQ     10$             ;Y,N-10$
034526' 032713 000040           BIT     #SETDED,(R3)    ;PRT DEV STILL DEDICATED TO THIS PROG?
034532' 001424                  BEQ     CPGNXT          ;Y,N-CPGNXT
034534' 042715 001000   10$:     BIC     #CAIDLE,(R5)    ;RESET THE 'ALL IDLE' FLAG
034540' 052715 002000           BIS     #CUSPGR,(R5)    ;SET 'USER PROG RUNNING' FLAG
034544' 032713 000030           BIT     #WT4IOT+CTPRIO,(R3) ;PROG WAITING FOR I/O TERM?
034550' 001430                  BEQ     CEXPRG          ;Y,N-CEXPRG
034552' 032713 000020           BIT     #CTPRIO,(R3)    ;THIS A CONSOLE OR PRINTER I/O?
034556' 001012                  BNE     CPGNXT          ;N,Y-CPGNXT
```

```
8691  034560' 032763 000040 000002          BIT     #DOIOT,POPSW(R3)        ;DO I/O TIME OUT?
8692  034566' 001006                        BNE     CPGNXT                  ;Y,N-CPGNXT
8693  034570' 005767 177672                 TST     CTOCK                   ;CHECK T/O THIS LOOP?
8694  034574' 001003                        BNE     CPGNXT                  ;Y,N-CPGNXT
8695  034576' 005363 000030                 DEC     PTOCNT(R3)              ;DECR TIME OUT CNT BY 1
8696  034602' 001455                        BEQ     CPGTOE                  ;TIMED OUT? (N,Y-CPGTOE)
8697  034604' 005724                 CPGNXT: TST     (R4)+                   ;INCR TO NXT CNTR ROUT TBL ENTRY
8698  034606' 022714 177777                 CMP     #177777,(R4)            ;END OF THE TABLE?
8699  034612' 001326                        BNE     CPGTTE                  ;Y,N-CPGTTE
8700  034614' 032715 001000                 BIT     #CAIDLE,(R5)            ;'ALL IDLE' FLAG SET?
8701  034620' 001705                        BEQ     CPGDSP                  ;Y,N-CPGDSP
8702  034622' 042715 002000                 BIC     #CUSPGR,(R5)            ;RESET 'USER PROG RUNNING' FLAG
8703  034626' 000167 143516                 JMP     CFCMND                  ;GO ASK FOR A CMND ENTRY
8704
8705                  001                     .IF DF MM
8706                                          .IFF
8707  034632' 062703 000124          CEXPRG: ADD     #PSTKCT,R3              ;GET ADR OF STACK STORAGE CNT
8708                                          .IFT
8709                                  CEXPRG: MOV     R3,R0                   ;POINT AT USER'S PAR'S STORAGE
8710                                          ADD     #PUPARS,R0
8711                                          MOV     #UPAR0,R1               ;SET UP PAR & PDR REG'S ADR'S
8712                                          MOV     #UPDR0,R2
8713                                          MOV     #8.,R5                  ;SET UP REG CNT
8714                                  12$:    MOV     16.(R0),(R2)+           ;LOAD USER PDR
8715                                          MOV     (R0)+,(R1)+             ;LOAD USER PAR
8716                                          DEC     R5                      ;DONE ALL REGS?
8717                                          BNE     12$                     ;Y,N-12$
8718                                          ADD     #PSTKCT,R3              ;GET ADR OF STK STORAGE CNT
8719                  000                     .ENDC
8720  034636' 020627 001200                  CMP     SP,#STACK               ;**** STACK POINTER CORRECT?
8721  034642' 001401                        BEQ     15$                     ;**** N,Y-15$
8722  034644' 000000                        HALT                            ;****
8723  034646' 012300                  15$:  MOV     (R3)+,R0                ;GET # OF WORDS SAVED OFF OF THE STACK
8724  034650' 060300                        ADD     R3,R0                   ;POINT PAST LAST WORD
8725                  001                     .IF DF MM
8726                                          .IFF
8727  034652' 020300                  20$:  CMP     R3,R0                   ;LOADED ALL WORDS ONTO STACK?
8728  034654' 001402                        BEQ     30$                     ;N,Y-30$
8729  034656' 014046                        MOV     -(R0),-(SP)             ;MOVE PROG'S WORD ONTO STACK
8730  034660' 000774                        BR      20$                     ;GO CK IF DONE
8731  034662' 005040                  30$:  CLR     -(R0)                   ;CLEAR OLD STACK COUNT
8732  034664' 011427                        MOV     (R4),(PC)+              ;SAVE CURR PROG TBL ADR
8733  034666' 000000                  CPGCPT: .WORD  0                       ;CURRENT PROG TABLE ADR
8734  034670' 010305                        MOV     R3,R5                   ;MOVE PRG TBL PNTR TO WK REG
8735  034672' 062705 000074                 ADD     #PSVREG-PSTKSV,R5       ;POINT AT SAVED REG'S
8736  034676' 012500                        MOV     (R5)+,R0                ;LOAD USER REGISTERS
8737  034700' 012501                        MOV     (R5)+,R1
8738  034702' 012502                        MOV     (R5)+,R2
8739  034704' 012503                        MOV     (R5)+,R3
8740  034706' 012504                        MOV     (R5)+,R4
8741  034710' 012546                        MOV     (R5)+,-(SP)
8742  034712' 011505                        MOV     (R5),R5
8743                                          .IFT
8744                                          MOV     PC,R1                   ;SET UP ADR OF USER STACK AREA
8745                                          ADD     #USRSTK-.,R1
8746                                          MOV     #USRSTK-2-PG6BA+P6CONS,USERR6 ;SET UP STK'S VIRTUAL ADR
```

```
8747                                    20$:    CMP     R3,R0                   ;LOADED ALL WORDS ON STK?
8748                                            BEQ     30$                     ;N,Y-30$
8749                                            MOV     -(R0),-(R1)             ;MOVE WORD ONTO USER STK
8750                                            SUB     #2,USERR6               ;UPDATE USER STK PNTR
8751                                            BR      20$                     ;GO CK IF DONE
8752                                    30$:    CLR     -(R0)                   ;CLEAR OLD STK CNT
8753                                            MOV     (R4),(PC)+              ;SAVE CURR PROG TBL ADR
8754                                    CPGCPT: .WORD   0                       ;CURRENT PROG TABLE ADR
8755                                            MOV     R3,R5                   ;MOVE PROG TBL PNTR TO WK REG
8756                                            ADD     #PSVREG-PSTKSV,R5       ;POINT AT SAVED REG'S
8757                                            MOV     10.(R5),-(R1)           ;PUT USER'S R5 ON HIS STACK
8758                                            MOV     (R5)+,R0                ;LOAD USER REGISTERS 0 - 4
8759                                            MOV     (R5)+,R1
8760                                            MOV     (R5)+,R2
8761                                            MOV     (R5)+,R3
8762                                            MOV     (R5)+,R4
8763                                            TST     (R5)+                   ;BYPASS ALREADY STORED R5
8764                                            MOV     (R5),R5                 ;LOAD USER'S PC INTO R5
8765                                            MOV     #USRPSW,-(SP)           ;LOAD USER MODE PSW ON STK
8766                                            MOV     #USMENT-PG6BA+P6CONS,-(SP) ;PUT ENTRY POINT ADR ON STK
8767                                            RTI                             ;GO INTO USER MODE IN-LINE
8768                                    USMENT: MOV     (PC)+,R6                ;LOAD USER'S STK PNTR
8769                                    USERR6: .WORD   0
8770                                            .IFTF
8771  034714' 005767  177570                    TST     TOEADR                  ;IS THIS FOR T/O ERR ROUT?
8772  034720' 001001                            BNE     40$                     ;N,Y-40$
8773  034722' 000205                            RTS     R5                      ;GO TO USER'S PROG
8774
8775  034724' 004577  177560            40$:    JSR     R5,@TOEADR              ;GO TO USER'S T/O ERR ROUT
8776  034730' 004567  000270                    JSR     R5,CPGRSV               ;SAVE PROG'S INFO
8777  034734' 000516                            BR      CKEX                    ;GO SET UP STANDARD ADRS
8778             000                            .ENDC
8779
8780  034736' 010300                    CPGTOE: MOV     R3,R0                   ;SET UP ADR OF T/O ROUT ADR
8781  034740' 062700  000306                    ADD     #PTLGTH+DTOEAD,R0
8782  034744' 005710                            TST     (R0)                    ;THIS DEV HAVE TIMEOUT ROUT?
8783  034746' 001716                            BEQ     CPGNXT                  ;Y,N-CPGNXT
8784  034750' 061000                            ADD     (R0),R0                 ;MAKE ITS ADR ABS
8785             001                            .IF DF MM
8786                                            BIC     #P4CONS,R0              ;ADJ ADR TO USER SPACE
8787             000                            .ENDC
8788  034752' 010067  177532                    MOV     R0,TOEADR               ;STORE T/O ERR ROUT ADR
8789  034756' 000725                            BR      CEXPRG                  ;GO TO USER PROG
```

```
                                .SBTTL  USER PROGRAM RETURN ROUTINES

                        ;*************************************************************************
                        ;
                        ;  THE RETURN ROUTINES PROVIDE VARIOUS ENTRY POINTS WHICH WILL SUSPEND
                        ;  USER PROGRAM EXECUTION FOR EITHER ONE POLLING LOOP, UNTIL AN
                        ;  I/O COMPLETES, OR INDEFINITELY. ALSO PROVIDED ARE END OF PROGRAM
                        ;  AND ERROR DETECTED ENTRY POINTS. THESE ENTRY POINTS ARE AS
                        ;  FOLLOWS:
                        ;
                        ;       CUPCR   =       USER PROGRAM CONTROL RELEASE
                        ;       CIOBSY  =       USER I/O DEVICE IS ALREADY BUSY AT INITIATION
                        ;       CUPGEX  =       END OF USER PROGRAM
                        ;       CUPGER  =       THE USER PROGRAM'S DEVICE ROUTINE
                        ;                       HAS DETECTED AN ERROR.
                        ;
                        ;*************************************************************************

                                ;USER PROGRAM ERROR RETURN POINT

034760' 004567  000240          CUPGER: JSR     R5,CPGRSV               ;SAVE REG'S & SET UP ADR'S
034764' 052713  000004                  BIS     #ERSTOP,(R3)            ;SET ERROR STOP FLAG
034770' 032763  100000  000002          BIT     #STONER,POPSW(R3)       ;STOP ON ERROR IN OPSW?
034776' 001075                          BNE     CKEX                    ;N,Y-CKEX
035000' 042713  000004                  BIC     #ERSTOP,(R3)            ;RESET ERROR STOP IN FLGWD
035004' 032763  001000  000002          BIT     #GTNXTD,POPSW(R3)       ;GO TO NEXT DEVICE ON ERROR?
035012' 001467                          BEQ     CKEX                    ;Y,N-CKEX
035014' 000406                          BR      CKPDVP                  ;GO SET UP FOR NEXT DEVICE


                                ;USER PROGRAM TERMINATION RETURN POINT


035016' 004567  000202          CUPGEX: JSR     R5,CPGRSV               ;SAVE REG'S AND SET UP ADR'S
035022' 032763  040000  000002          BIT     #CYCPRG,POPSW(R3)       ;CYCLE PROG ON CURR DEV?
035030' 001043                          BNE     CKCLRG                  ;N,Y-CKCLRG
035032' 126327  000034  000017  CKPDVP: CMPB    PDPNTR(R3),#15.         ;DEVICE # PNTR AT LAST DEV?
035040' 103430                          BLO     30$                     ;Y,N-30$
035042' 032763  002000  000002  20$:    BIT     #CYCDVL,POPSW(R3)       ;CYCLE DEVICE LIST?
035050' 001403                          BEQ     22$                     ;Y,N-22$
035052' 105063  000034                  CLRB    PDPNTR(R3)              ;SET DEV PNTR TO FIRST DEV
035056' 000430                          BR      CKCLRG                  ;GO RESTART AT FIRST DEV
035060' 032763  000020  000002  22$:    BIT     #AUTORP,POPSW(R3)       ;DISPLAY COUNTS AUTOMATICALLY?
035066' 001002                          BNE     24$                     ;Y,N-24$
035070' 004767  000402                  JSR     PC,CRPTEP               ;GO ISSUE THE REPORT
035074' 032763  000001  000002  24$:    BIT     #NOCOMP,POPSW(R3)       ;PRINT PROG COMPLETED MSG?
035102' 001004                          BNE     28$                     ;Y,N-28$
035104' 004567  000512                  JSR     R5,ULISTM               ;ISSUE PROG XX COMPLETED MSG
035110' 000456                          .WORD   PGCOMP-.
035112' 177767                          .WORD   -9.
035114' 042713  100000          28$:    BIC     #ACTIVE,(R3)            ;RESET THIS PROG'S ACTIVE FLAG
035120' 000424                          BR      CKEX                    ;GO TO ROUTINE EXIT
035122' 105263  000034          30$:    INCB    PDPNTR(R3)              ;INCR DEV # PNTR
035126' 004767  000236                  JSR     PC,CPGCDN               ;POINT AT NEXT DEV #
```

```
8847  035132' 122712  000377                  CMPB    #377,(R2)                ;IS IT A DEV #?
8848  035136' 001741                          BEQ     20$                      ;Y,N-20$
8849  035140' 032763  000030  000002  CKCLRG: BIT     #AUTORP+AURPEP,POPSW(R3)  ;DO AUTO DISPLAY OF COUNTS?
8850  035146' 001002                          BNE     40$                      ;Y,N-40$
8851  035150' 004767  000322                  JSR     PC,CRPTEP                ;GO ISSUE THE REPORT
8852  035154' 004767  000210          40$:    JSR     PC,CPGCDN                ;POINT AT NEXT DEV #
8853  035160' 111263  000035                  MOVB    (R2),PCURDV(R3)          ;STORE IT AS CURR DEV #
8854  035164' 004567  000214                  JSR     R5,CPGHSK                ;GO DO PROG HOUSEKEEPING
8855  035170' 000000                          .WORD   0                        ;  HSKP CNTS PER OPSW FLAG
8856              001                         .IF DF MM
8857                                          .IFT
8858                                          MOV     PC,R5                    ;SET THE REMAP FLAG
8859                                          BR      CKEX1                    ;GO SET UP ADRS
8860
8861                                  CKEX:   CLR     R5                       ;RESET THE REMAP FLAG
8862                                  CKEX1:  TRAP    2                        ;GO INTO KERNEL MODE IN-LINE
8863                                  UPGRET: ADD     #4,SP                    ;REMOVE TRAP'S PSW & PC
8864                                          MOV     CPGCPE,R4                ;GET CURR CNTRL ROUT TBL ADR
8865                                          MOV     (R4),@#KPAR4             ;MAP PAR 4 TO THIS USER
8866                                          MOV     #P4CONS,R3               ;SET UP HIS VIRTUAL ADR
8867                                          TST     R5                       ;REMAP HIS PAR/PDR STORAGE
8868                                          BEQ     50$                      ;Y,N-50$
8869                                          JSR     PC,MAPUPG                ;MAP PAR/PDR STORAGE
8870                                  50$:    MOV     PC,R5                    ;SET UP SYSTEM FLGWD ADR
8871                                          ADD     #CSYSFW-.,R5
8872                                          .IFF
8873  035172' 016704  177274          CKEX:   MOV     CPGCPE,R4                ;GET CURR CNTRL ROUT TBL ADR
8874  035176' 010705                          MOV     PC,R5                    ;SET UP SYSTEM FLGWD ADR
8875  035200' 062705  177014                  ADD     #CSYSFW-.,R5
8876              000                         .ENDC
8877  035204' 005067  177456                  CLR     CPGCPT                   ;CLEAR CURR PROG TBL ADR
8878  035210' 000167  177370                  JMP     CPGNXT                   ;GO CHECK NEXT PROG ENTRY
8879
8880
8881                                          ;USER PROGRAM I/O BUSY, CONTROL
8882                                          ;RELEASE, AND KILL RETURN POINTS
8883
8884
8885              001                         .IF DF MM
8886                                  CKILEK: MOV     #STACK,R6                ;RESTORE KERNEL'S STK PNTR
8887                                          MOV     #USRPSW,-(SP)            ;LOAD USER MODE PSW ON STK
8888                                          MOV     #CKILEX-PG6BA+P6CONS,-(SP) ;SET UP KILL ABORT ADR
8889                                          RTI                              ;GO INTO USER MODE FOR THE KILL
8890              000                         .ENDC
8891  035214' 005746                  CKILEX: TST     -(SP)                    ;ADD A DUMMY WORD ON THE STACK
8892
8893  035216'                         CUPCR:
8894  035216' 004567  000002          CIOBSY: JSR     R5,CPGRSV                ;SAVE REG'S & SET UP ADR'S
8895  035222' 000763                          BR      CKEX                     ;GO CHECK NXT PROG SLOT
8896
8897
8898                                  ;       **************************************************
8899                                  ;       *                                                *
8900                                  ;       *  USER PROGRAM DISPATCHER & RETURN SUBROUTINES  *
8901                                  ;       *                                                *
8902                                  ;       **************************************************
```

```
                                        ;SAVE USER REGISTERS & STACK INFO
                                        ;JSR    R5,CPGRSV       S/R CALL
035224' 010527          CPGRSV: MOV     R5,(PC)+                ;SAVE RETURN ADR
035226' 000000          CPSVR5: .WORD   0
            001                 .IF DF MM
                                .IFF
035230' 016705  177432          MOV     CPGCPT,R5               ;GET CURRENT PROG TBL ADR
                                .IFT
                                CLR     R5                      ;GET VIRTUAL PROG TBL ADR
            000                 .ENDC
035234' 062705  000236          ADD     #PUSRPC,R5              ;POINT AT PC STORAGE AREA
035240' 012615                  MOV     (SP)+,(R5)              ;SAVE PROG'S PC
035242' 012645                  MOV     (SP)+,-(R5)             ;SAVE PROG'S R5
035244' 010445                  MOV     R4,-(R5)                ;SAVE R4 THRU R0
035246' 010345                  MOV     R3,-(R5)
035250' 010245                  MOV     R2,-(R5)
035252' 010145                  MOV     R1,-(R5)
035254' 010045                  MOV     R0,-(R5)
035256' 010503                  MOV     R5,R3                   ;PROG TBL PNTR TO R3
035260' 162703  000222          SUB     #PSVREG,R3
            001                 .IF DF MM
                                .IFF
035264' 012700  001200          MOV     #STACK,R0               ;SET UP STACK PNTR MAX VALUE
                                .IFT
                                MOV     #USRSTK-PG6BA+P6CONS,R0 ;SET UP STK PNTR MAX VALUE
            000                 .ENDC
035270' 160600                  SUB     SP,R0                   ;GET # OF WORDS ON STACK
035272' 100015                  BPL     70$                     ;STACK ADR SCREWED UP? (Y,N-70$)
035274' 012767  000212  000012  MOV     #INVSTK-60$,60$         ;STORE INV STK ERR MSG ADR
035302' 010645          50$:    MOV     SP,-(R5)                ;SAVE BAD STK PNTR
            001                 .IF DF MM
                                .IFF
035304' 012706  001200          MOV     #STACK,R6               ;REINITIALIZE STK PNTR
                                .IFT
                                MOV     #USRSTK-PG6BA+P6CONS,R6 ;REINITIALIZE STK PNTR
            000                 .ENDC
035310' 004567  000306          JSR     R5,ULISTM               ;ISSUE "INV STK ADR" ERROR MSG OR
035314' 000000          60$:    .WORD   XXXX
035316' 177760                  .WORD   -16.
035320' 042713  100000          BIC     #ACTIVE,(R3)            ;RESET THIS PROG'S ACTIVE FLAG
035324' 000722                  BR      CKEX                    ;GO CK FOR ANOTHER USER PROG
035326' 020027  000074  70$:    CMP     R0,#PSVREG-PSTKSV       ;TOO MANY WORDS ON STACK?
035332' 101404                  BLOS    80$                     ;Y,N-80$
035334' 012767  000232  177752  MOV     #STKBIG-60$,60$         ;STORE STK TOO BIG ERR MSG ADR
035342' 000757                  BR      50$                     ;GO SHUT DOWN THIS PROG
035344' 162705  000076  80$:    SUB     #PSVREG-PSTKCT,R5       ;POINT TO STACK CNT STORAGE
035350' 010025                  MOV     R0,(R5)+                ;STORE STACK COUNT
035352' 162700  000002  90$:    SUB     #2,R0                   ;DECR STACK WORD CNT
035356' 100402                  BMI     100$                    ;ALL WORDS OFF STACK? (N,Y-100$)
035360' 012625                  MOV     (SP)+,(R5)+             ;STORE WORD OFF STACK
035362' 000773                  BR      90$                     ;GO DECR CNT
            001                 .IF DF MM
```

```
                                        .IFF
035364' 016707  177636          100$:   MOV     CPSVR5,PC               ;EXIT TO IN-LINE ADR
                                        .IFT
                                100$:   MOV     R3,R0                   ;GET ADR OF PAR STORAGE
                                        ADD     #PUPARS,R0
                                        MOV     #UPAR0,R1               ;GET ADRS OF USER'S MODE
                                        MOV     #UPDR0,R2               ;PAR & PDR REGS
                                        MOV     #8.,R5                  ;SET UP REG CNT
                                110$:   MOV     (R2)+,16.(R0)           ;STORE PDR VALUE
                                        MOV     (R1)+,(R0)+             ;STORE PAR VALUE
                                        DEC     R5                      ;DONE ALL REGS?
                                        BNE     110$                    ;Y,N-110$
                                        MOV     CPSVR5,PC               ;RETURN IN-LINE
                000                     .ENDC


                                        ;GET ADR IN R2 OF UNIT # POINTED TO BY "PDPNTR"

                                                ;JSR    PC,CPGCDN       S/R CALL
                                                ;R3 CONTAINS PROG TBL ADR
                                                ;DESTROYS R2

035370' 116302  000034          CPGCDN: MOVB    PDPNTR(R3),R2           ;GET DEV PNTR
035374' 060302                          ADD     R3,R2                   ;SET UP DEV # ADR
035376' 062702  000036                  ADD     #PDNUMS,R2
035402' 000207                          RTS     PC                      ;EXIT


                                        ;HOUSEKEEP PROG TBL & OPTIONALLY THE DEV ROUT

                                                ;JSR    R5,CPGHSK       S/R CALL
                                                ;.WORD  0 OR 1          0 = DO DEV ROUT HSKP PER OPSW
                                                ;                       1 = UNCOND. DO DEV ROUT HSKP
                                                ;R3 CONTAINS PROG TBL ADR
                                                ;DESTROYS R0,R1,R2

035404' 010300                  CPGHSK: MOV     R3,R0                   ;PROG TBL ADR TO WK REG
035406' 062700  000056                  ADD     #PTEMO,R0               ;POINT AT TMO0 WORD
035412' 012701  000070                  MOV     #PUSRPC-PTEMO/2,R1      ;GET # OF WORDS TO CLEAR
035416' 005020                  10$:    CLR     (R0)+                   ;CLEAR TMP, STACK, & REG
035420' 005301                          DEC     R1                      ;STORAGE FOR THIS PROG
035422' 001375                          BNE     10$
035424' 016320  000024                  MOV     POBJST(R3),(R0)+        ;SET PROG'S PC TO OBJ CODE START
035430' 062700  000004                  ADD     #PTSIZE-PUSRPC+DEVFWD,R0  ;POINT AT DEV ROUT FLGWD
035434' 012701  000011                  MOV     #DEVDRA-DEVFWD/2,R1     ;GET # OF DEV ROUT WDS TO CLEAR
035440' 005020                  20$:    CLR     (R0)+                   ;CLEAR DEV ROUT FLAGWORD &
035442' 005301                          DEC     R1                      ;INTERFACE WORDS
035444' 001375                          BNE     20$
035446' 012567  000020                  MOV     (R5)+,35$               ;STORE HOUSEKEEP FLAGWD
035452' 062700  000010          30$:    ADD     #DHKPAD-DEVDRA,R0       ;POINT AT DEV ROUT HSKP ROUT ADR
035456' 005710                          TST     (R0)                    ;IS THERE A HSKP ROUT?
035460' 001405                          BEQ     40$                     ;Y,N-40$
035462' 061000                          ADD     (R0),R0
035464' 016302  000002                  MOV     POPSW(R3),R2            ;SET UP PROG'S OPSW
035470' 004510                          JSR     R5,(R0)                 ;GO TO DEVICE'S HSKP ROUT
```

```
9015  035472' 000000                  35$:    .WORD   XXXX                    ;HOUSEKEEP FLAGWD
9016  035474' 000205                  40$:    RTS     R5                      ;EXIT IN-LINE
9017
9018
9019                                          ;GO TO DEVICE ROUTINE REPORT ROUTINE
9020
9021                                          ;JSR    PC,CGORPT       REPORT CMND S/R CALL
9022                                          ;         CRPTEP        END OF USER PROG PASS S/R CALL
9023                                          ;R3 CONTAINS PROG TBL ADR
9024                                          ;DESTROYS R0
9025
9026  035476' 012767  000001  000016  CRPTEP: MOV     #1,CRPFLG               ;SET UP END OF PASS FLGWD
9027  035504' 010300                  CGORPT: MOV     R3,R0                   ;POINT TO ADR OF REPORT ROUT
9028  035506' 062700  000300                  ADD     #PTLGTH+DERPAD,R0
9029  035512' 005710                          TST     (R0)                    ;IS THERE A ROUT ADR?
9030  035514' 001403                          BEQ     CGOREX                  ;Y,N-CGOREX
9031  035516' 061000                          ADD     (R0),R0                 ;SET UP ABS ADR OF REPORT ROUT
9032  035520' 004510                          JSR     R5,(R0)                 ;GO TO DEV ROUT ERROR REPORT S/R
9033  035522' 000000                  CRPFLG: .WORD   XXXX
9034  035524' 000207                  CGOREX: RTS     PC                      ;EXIT IN-LINE
9035
9036                                          .NLIST  BEX
9037  035526' 042452  025122  044440  INVSTK: .ASCII  /*ER* INV STK ADR/
9038  035546' 042452  025122  051440  STKBIG: .ASCII  /*ER* STK TOO BIG/
9039  035566' 047503  050115  042514  PGCOMP: .ASCII  /COMPLETED/
9040                                          .LIST   BEX
9041          035600'                         .EVEN
```

```
                                        .SBTTL  FORMATS FOR PROGRAM & DEVICE ROUTINE TABLES

                                ;       PROGRAM TABLE FORMAT

                                        .IF DF MM
                                        .IFF
000242                          PTLGTH= 162.    ;PROGRAM TABLE LENGTH
                                        .IFT
                                PTLGTH= 212.    ;PROGRAM TABLE LENGTH (MEM MGMNT VERSION)
                                        .ENDC

000000                          PFLGWD= +0.     ;PROGRAM FLAG WORD - 1 WORD

                                        URSTOP= 2       ; 1 = USER HAS STOPPED THIS PROGRAM
                                        ERSTOP= 4       ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
                                        WT4IOT= 10      ; 1 = WAITING FOR I/O TERMINATION
                                        CTPRIO= 20      ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
                                        SETDED= 40      ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
                                        OCPRES= 100     ; 1 = OBJ CODE IS PRESENT
                                        .IF DF MM
                                        USEUBM= 200     ; 1 = THIS PROG USES THE UNIBUS MAP
                                        .ENDC
                                        ACTIVE= 100000  ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

000002                          POPSW=  +2.     ;PROGRAM'S OPERATION SWITCHES - 1 WORD

                                        STONER= 100000  ; 1 = STOP PROG EXECUTION UPON ERROR
                                        CYCPRG= 40000   ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
                                        PRONER= 20000   ; 1 = DO NOT PRINT ON ERROR
                                        BIT12=  10000   ; 0 = NOT USED
                                        BIT11=  4000    ; 0 = NOT USED
                                        CYCDVL= 2000    ; 1 = CYCLE THE DEVICE LIST
                                        GTNXTD= 1000    ; 1 = CYCLE ON SAME DEVICE UPON ERROR
                                        DOERCK= 400     ; 1 = DON'T DO ERROR CHECKING
                                        SPOPER= 200     ; 1 = DEVICE SPECIAL OPERATION
                                        BIT6=   100     ; 0 = NOT USED
                                        DOIOT=  40      ; 1 = DO NOT PERFORM I/O TIMEOUT
                                        AUTORP= 20      ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
                                        AURPEP= 10      ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
                                        HSKPEP= 4       ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
                                        PFBBOV= 2       ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
                                        NOCOMP= 1       ; 1 = DO NOT PRINT PROG COMPLETED MSG

000004                  PFWADR= +4.     ;*;PROGRAM FLAGWORD ADDRESS - 1 WORD

000006                  PASCIN= +6.     ;PROGRAM'S NUMBER IN ASCII - 1 WORD

000010                  PNAME=  +8.     ;PROGRAM'S NAME IN ASCII - 6 BYTES

000016                  PRDIOA= +14.    ;ADDRESS OF READ I/O AREA - 1 WORD

000020                  PWRIOA= +16.    ;ADDRESS OF WRITE I/O AREA - 1 WORD

000022                  PSRCST= +18.    ;SOURCE STATEMENTS START ADDRESS - 1 WORD

000024                  POBJST= +20.    ;OBJECT CODE START ADDRESS - 1 WORD
```

```
9099
9100    000026                  PLNGTH= +22.    ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD
9101
9102    000030                  PTOCNT= +24.    ;I/O TIMEOUT COUNT - 1 WORD
9103
9104    000032                  PMDLCD= +26.    ;DEV ROUT MODEL # CODE - 1 WORD
9105
9106    000034                  PDPNTR= +28.    ;CURRENT DEVICE NUMBER POINTER - 1 BYTE
9107
9108    000035                  PCURDV= +29.    ;CURRENT DEVICE # - 1 BYTE
9109
9110    000036                  PDNUMS= +30.    ;DEVICE NUMBERS - 16 BYTES
9111
9112    000056                  PTEM0=  +46.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9113
9114    000060                  PTEM1=  +48.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9115
9116    000062                  PTEM2=  +50.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9117
9118    000064                  PTEM3=  +52.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9119
9120    000066                  PTEM4=  +54.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9121
9122    000070                  PTEM5=  +56.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9123
9124    000072                  PTEM6=  +58.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9125
9126    000074                  PTEM7=  +60.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9127
9128    000076                  PTEM8=  +62.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9129
9130    000100                  PTEM9=  +64.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9131
9132    000102                  PTEM10= +66.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9133
9134    000104                  PTEM11= +68.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9135
9136    000106                  PTEM12= +70.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9137
9138    000110                  PTEM13= +72.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9139
9140    000112                  PTEM14= +74.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9141
9142    000114                  PTEM15= +76.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
9143
9144    000116                  PNBR=   +78.    ;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
9145
9146    000120                  PSRC=   +80.    ;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
9147
9148    000122                  PDST=   +82.    ;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
9149
9150    000124                  PSTKCT= +84.    ;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
9151
9152    000126                  PSTKSV= +86.    ;STACK WORDS STORAGE AREA - 30 WORDS
9153
9154    000222                  PSVREG= +146.   ;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
```

```
9155
9156  000236                    PUSRPC= +158.   ;USER'S CURRENT PROGRAM COUNTER - 1 WORD
9157
9158  001                               .IF DF MM
9159                              .IFF
9160  000240                    PTSIZE= +160.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD
9161
9162  000242                    PTEND=  +162.   ;END OF PROGRAM TABLE
9163                              .IFT
9164                            PRDIOX= +160.   ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS
9165
9166                            PRDIOV= +164.   ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS
9167
9168                            PWRIOX= +168.   ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS
9169
9170                            PWRIOV= +172.   ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS
9171
9172                            PUPARS= +176.   ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS
9173
9174                            PUPDRS= +192.   ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS
9175
9176                            PUBMAP= +208.   ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD
9177
9178                            PTSIZE= +210.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD (MEM MGMNT VERSION)
9179
9180                            PTEND=  +212.   ;END OF PROGRAM TABLE (MEM MGMNT VERSION)
9181  000                               .ENDC
```

```
                                ;       DEVICE ROUTINE TABLE

000116                          DRTLTH= 78.     ;DEVICE ROUTINE TABLE LENGTH
                                ;
000000                          DEVRSZ= +0.     ;DEVICE ROUTINE SIZE IN BYTES - 1 WORD
000002                          DEVFWD= +2.     ;DEVICE ROUTINE FLAGWORD - 1 WORD
000004                          DEVIW1= +4.     ;DEVICE INTERFACE WORD # 1 - 1 WORD
000006                          DEVIW2= +6.     ;DEVICE INTERFACE WORD # 2 - 1 WORD
000010                          DEVIW3= +8.     ;DEVICE INTERFACE WORD # 3 - 1 WORD
000012                          DEVIW4= +10.    ;DEVICE INTERFACE WORD # 4 - 1 WORD
000014                          DEVIW5= +12.    ;DEVICE INTERFACE WORD # 5 - 1 WORD
000016                          DEVIW6= +14.    ;DEVICE INTERFACE WORD # 6 - 1 WORD
000020                          DEVIW7= +16.    ;DEVICE INTERFACE WORD # 7 - 1 WORD  (SIZE)
000022                          DEVIW8= +18.    ;DEVICE INTERFACE WORD # 8 - 1 WORD  (ERR)
000024                          DEVDRA= +20.    ;DEVICE REGISTERS ADDRESS - 1 WORD
000026                          DEVIVA= +22.    ;DEVICE INTERRUPT VECTOR ADDRESS - 1 WORD
000030                          DEVRPS= +24.    ;DEVICE READ PROCESSOR STATUS WORD (BUS REQ) - 1 WORD
000032                          DEVWPS= +26.    ;DEVICE WRITE PROC STATUS WORD (BUS REQ) - 1 WORD
000034                          DHKPAD= +28.    ;DEVICE ROUT HOUSEKEEPING ROUT REL ENTRY ADR - 1 WORD
000036                          DERPAD= +30.    ;DEVICE ROUT REPORT ROUT REL ENTRY ADR - 1 WORD
000040                          DKILAD= +32.    ;DEVICE ROUT KILL ROUTINE REL ENTRY ADR - 1 WORD
000042                          DECTAD= +34.    ;DEVICE ROUT ERROR COUNTER REL ADR - 1 WORD
000044                          DTOEAD= +36.    ;DEVICE ROUT TIMEOUT ERR ROUT REL ENTRY ADR - 1 WORD
000046                          DEVIOB= +38.    ;DEVICE I/O BUSY BRANCH ADDRESS (CIOBSY) - 1 WORD
000050                          DEVDER= +40.    ;DEVICE ERROR BRANCH ADDRESS (CUPGER) - 1 WORD
000052                          DVUPRT= +42.    ;USER MODE PRINT BRANCH ADDRESS (ULIST) - 1 WORD
000054                          DVCPRT= +44.    ;CMND MODE PRINT BRANCH ADDRESS (CLIST) - 1 WORD
000056                          DEVBTA= +46.    ;CONVERT BINARY TO ASCII BR ADR (BINASC) - 1 WORD
000060                          DVBTDA= +48.    ;CONVERT BINARY TO DECIMAL ASCII BR ADR (BTASLZ) - 1 WORD
```

```
9239            000062                  DVPDTA= +50.    ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
9240
9241            000064                  DVSFWD= +52.    ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
9242
9243            000066                  DVSVEC= +54.    ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
9244
9245            000070                  DVCVEC= +56.    ;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
9246
9247            000072                  DVTVEC= +58.    ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
9248
9249            000074                  DVRINT= +60.    ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
9250
9251            000076                  DVGETB= +62.    ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
9252
9253            000100                  DVPUTB= +64.    ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
9254
9255            000102                  DEVSTP= +66.    ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
9256
9257            000104                  DEVETP= +68.    ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
9258
9259            000106                  DVPTEP= +70.    ;PACK TABLE EXTEN. REL POINTER - 1 WORD
9260
9261            000110                  DVVTEP= +72.    ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
9262
9263            000112                  DVCTEP= +74.    ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
9264
9265            000114                  DVIWSP= +76.    ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
9266
9267            000116                  DRTEND= +78.    ;END OF DEVICE ROUTINE TABLE
```

```
                                        .SBTTL  LIST CONTROL ROUTINES

                                ;*****************************************************************
                                ;
                                ;  THIS ROUTINE DETERMINES IF A LIST DEVICE HAS BEEN ASSIGNED
                                ;  AND THEN BRANCHES TO EITHER THE CONSOLE TERMINAL ROUTINE
                                ;  OR TO THE PRINTER ROUTINE.
                                ;
                                ;  LINKAGE:     JSR     R5,CLIST        MPG LIST
                                ;               .WORD   TAG-.           REL ADR OF I/O AREA
                                ;               .WORD   COUNT           # OF CHARS TO TRANSFER
                                ;
                                ;  EXITS TO EITHER: CTWR OR CPRT
                                ;
                                ;*****************************************************************

035600' 016700  142210          CLIST:  MOV     CLISTI,R0               ;GET ADR OF LIST DEV
035604' 005760  000004                  TST     4(R0)                   ;IS THERE AN ALT LIST DEV?
035610' 001002                          BNE     10$                     ;N,Y-10$
035612' 000177  142222                  JMP     @CTWR                   ;GO TO MPG'S CONS TERM WRITE
035616' 000177  142176          10$:    JMP     @CPRT                   ;GO TO MPG'S PRINT

                                        ;USER LIST S/R FOR DEVICE ROUTINES

                                        ;JSR    R5,ULIST        S/R CALL
                                        ;.WORD  ADR             ABS DATA ADR
                                        ;.WORD  CNT             BYTE COUNT

035622' 004567  002770          ULISTM: JSR     R5,LTSTDE               ;GO TEST DEDICATED FLAGS
035626' 010567  000040                  MOV     R5,UPRADR               ;GET REL MSG ADR & MAKE
035632' 062567  000034                  ADD     (R5)+,UPRADR            ;IT ABS
035636' 000404                          BR      ULSTCM                  ;GO TO COMMON POINT

035640' 004567  002752          ULIST:  JSR     R5,LTSTDE               ;GO TEST DEDICATED FLAGS
035644' 012567  000022                  MOV     (R5)+,UPRADR            ;GET ABS DATA ADR
035650' 011567  000020          ULSTCM: MOV     (R5),UPRBYC             ;GET BYTE COUNT
035654' 100402                          BMI     10$                     ;CNT NEG? (N,Y-10$)
035656' 005467  000012                  NEG     UPRBYC                  ;NEGATE THE BYTE COUNT
035662' 105067  002005          10$:    CLRB    LPRRDE                  ;RESET REALLY DEDICATED FLAG
035666' 004567  002134                  JSR     R5,UPRASC               ;ISSUE MSG USING PRINT ASCII ROUT
035672' 000000          UPRADR: .WORD   XXXX
035674' 000000          UPRBYC: .WORD   XXXX
035676' 005725                  TST     (R5)+                           ;WAS SUPPLIED BYTE CNT NEG?
035700' 100403                  BMI     20$                             ;N,Y-20$
035702' 004567  002404                  JSR     R5,UPCRLF               ;ISSUE <CR> & <LF>
035706' 000404                          BR      30$                     ;GO TO EXIT
035710' 105067  001757          20$:    CLRB    LPRRDE                  ;RESET REALLY DEDICATED FLAG
035714' 004567  002364                  JSR     R5,ULCRLF               ;ISSUE <CR> & <LF> FOR LAST MSG
035720' 000205          30$:    RTS     R5                              ;EXIT TO USER PROG
```

```
                                        .SBTTL  SUBROUTINES USED BY USER PROGRAM STATEMENTS
                                ;
                                ;
                                ;       ADD MULTIPLE SUBROUTINE
                                ;
035722' 012500          LADDM:  MOV     (R5)+,R0        ;GET COUNT
035724' 012501                  MOV     (R5)+,R1        ;FIRST VARIABLE POINTER
035726' 012502                  MOV     (R5)+,R2        ;SECOND VARIABLE POINTER
035730' 060001                  ADD     R0,R1           ;POINTERS
035732' 060002                  ADD     R0,R2           ;TO LSB
035734' 005067  000056          CLR     LCARRY          ;CLEAR SOFTWARE
035740' 000241                  CLC                     ;AND HARDWARE CARRY BITS
035742' 162701  000002          SUB     #2,R1
035746' 162702  000002          SUB     #2,R2
035752' 111103          LADDML: MOVB    (R1),R3         ;GET
035754' 111204                  MOVB    (R2),R4         ;OPERANDS (ONE BYTE)
035756' 066704  000034          ADD     LCARRY,R4       ;AND PREVIOUS CARRY
035762' 103001                  BCC     LADDCC          ;IF NO CARRY CLEAR SOFT CARRY
035764' 000402                  BR      LADD
035766' 005067  000024  LADDCC: CLR     LCARRY
035772' 060304          LADD:   ADD     R3,R4           ;ADD TWO OPERANDS
035774' 005567  000016          ADC     LCARRY          ;ADD ANY CARRY TO SOFT CARRY
036000' 110412                  MOVB    R4,(R2)         ;PUT BACK RESULT OPERAND
036002' 005300                  DEC     R0
036004' 001403                  BEQ     LADDEX          ;EXIT WHEN DONE
036006' 004567  000102          JSR     R5,BUMP         ;GET NEXT MSB LIKE REAL COMPUTER
036012' 000757                  BR      LADDML
036014' 000205          LADDEX: RTS     R5              ;RETURN
036016' 000000          LCARRY: .WORD   0
                                ;
                                ;       SUBTRACT MULTIPLE SUBROUTINE
                                ;
036020' 012500          LSUBM:  MOV     (R5)+,R0        ;GET COUNT
036022' 012501                  MOV     (R5)+,R1        ;FIRST VARIABLE POINTER
036024' 012502                  MOV     (R5)+,R2        ;SECOND VARIABLE POINTER
036026' 060001                  ADD     R0,R1           ;POINTERS
036030' 060002                  ADD     R0,R2           ;TO LSB
036032' 005067  177760          CLR     LCARRY          ;CLEAR SOFTWARE
036036' 000241                  CLC                     ;AND HARDWARE CARRIES
036040' 162701  000002          SUB     #2,R1
036044' 162702  000002          SUB     #2,R2
036050' 111103          LSUBML: MOVB    (R1),R3         ;GET
036052' 111204                  MOVB    (R2),R4         ;OPERANDS (ONE BYTE)
036054' 166704  177736          SUB     LCARRY,R4       ;SUBTRACT PREVIOUS CARRY
036060' 103001                  BCC     LSUBCC          ;IF NO CARRY CLEAR SOFT CARRY
036062' 000402                  BR      LSUB
036064' 005067  177726  LSUBCC: CLR     LCARRY
036070' 160304          LSUB:   SUB     R3,R4           ;SUBTRACT TWO OPERANDS
036072' 005567  177720          ADC     LCARRY          ;REMEMBER ANY CARRY
036076' 110412                  MOVB    R4,(R2)         ;PUT BACK RESULT OPERAND
036100' 005300                  DEC     R0
036102' 001403                  BEQ     LSUBEX          ;EXIT WHEN DONE
036104' 004567  000004          JSR     R5,BUMP         ;GET NEXT MSB LIKE REAL COMPUTER
036110' 000757                  BR      LSUBML
036112' 000205          LSUBEX: RTS     R5              ;RETURN
```

```
9377                                    ;
9378                                    ;       BUMP R0 AND R1 TO NEXT MSB SUBROUTINE
9379                                    ;
9380  036114' 032702  000001    BUMP:   BIT     #1,R2                   ;IF BIT 0 RESET
9381  036120' 001403                    BEQ     LBUMP1                  ;GO ADD ONE
9382  036122' 162702  000003            SUB     #3,R2                   ;OTHERWISE SUBTRACT 3
9383  036126' 000401                    BR      LBUMP2
9384  036130' 005202            LBUMP1: INC     R2
9385  036132' 032701  000001    LBUMP2: BIT     #1,R1                   ;IF BIT 0 RESET
9386  036136' 001403                    BEQ     LBUMP3                  ;GO ADD ONE
9387  036140' 162701  000003            SUB     #3,R1                   ;OTHERWISE SUBTRACT 3
9388  036144' 000401                    BR      LBUMP4
9389  036146' 005201            LBUMP3: INC     R1
9390  036150' 000205            LBUMP4: RTS     R5                      ;RETURN
9391                                    ;
9392                                    ;       FILL WITH VARIABLE SUBROUTINE
9393                                    ;
9394  036152' 012504            LFILV:  MOV     (R5)+,R4                ;GET ADDRESS
9395  036154' 012503                    MOV     (R5)+,R3                ;BYTE COUNT
9396  036156' 012502                    MOV     (R5)+,R2                ;AND VARIABLE
9397  036160' 032704  000001    LFVOET: BIT     #1,R4                   ;IS ADDRESS ODD?
9398  036164' 001403                    BEQ     LFVCBC                  ;NO-CHECK BYTE COUNT
9399  036166' 110224            LFVODD: MOVB    R2,(R4)+                ;MOVE ONE BYTE OF VAR
9400  036170' 005303                    DEC     R3                      ;DECREMENT BYTE COUNT
9401  036172' 001410                    BEQ     LFVEXT                  ;EXIT IF ZERO
9402  036174' 022703  000001    LFVCBC: CMP     #1,R3                   ;CHECK BYTE COUNT
9403  036200' 001772                    BEQ     LFVODD                  ;IF=1 TREAT AS ODD
9404  036202' 010224            LFVEVN: MOV     R2,(R4)+                ;MOVE ONE WORD OF VAR
9405  036204' 162703  000002            SUB     #2,R3                   ;DECREMENT BYTE COUNT
9406  036210' 001401                    BEQ     LFVEXT                  ;EXIT IF ZERO
9407  036212' 000770                    BR      LFVCBC                  ;OTHERWISE CHECK BYTE CNT
9408  036214' 000205            LFVEXT: RTS     R5                      ;RETURN
9409                                    ;
9410                                    ;       FILL WITH RANDOM SUBROUTINE
9411                                    ;
9412  036216' 012504            LFILR:  MOV     (R5)+,R4                ;GET ADDRESS
9413  036220' 012503                    MOV     (R5)+,R3                ;AND BYTE COUNT
9414  036222' 032704  000001    LFROET: BIT     #1,R4                   ;IS ADDRESS ODD?
9415  036226' 001406                    BEQ     LFRCBC                  ;NO-CHECK BYTE COUNT
9416  036230' 004567  003070    LFRODD: JSR     R5,LRANDM               ;GET RANDOM NBR
9417  036234' 116724  003174            MOVB    LHINUM,(R4)+            ;MOVE IN ONE BYTE
9418  036240' 005303                    DEC     R3                      ;DECREMENT BYTE COUNT
9419  036242' 001413                    BEQ     LFREXT                  ;EXIT IF ZERO
9420  036244' 022703  000001    LFRCBC: CMP     #1,R3                   ;CHECK BYTE COUNT
9421  036250' 001767                    BEQ     LFRODD                  ;IF=1 TREAT AS ODD
9422  036252' 004567  003046    LFREVN: JSR     R5,LRANDM               ;GET RANDOM NBR
9423  036256' 016724  003152            MOV     LHINUM,(R4)+            ;MOVE IN ONE WORD
9424  036262' 162703  000002            SUB     #2,R3                   ;DECREMENT BYTE COUNT
9425  036266' 001401                    BEQ     LFREXT                  ;EXIT IF ZERO
9426  036270' 000765                    BR      LFRCBC                  ;OTHERWISE CHECK BYTE CNT
9427  036272' 000205            LFREXT: RTS     R5                      ;RETURN
9428                                    ;
9429                                    ;       FILL WITH ASCII SUBROUTINE
9430                                    ;
9431  036274' 012504            LFILG:  MOV     (R5)+,R4                ;GET ADDRESS
9432  036276' 012503                    MOV     (R5)+,R3                ;AND BYTE COUNT
```

```
 9433  036300' 112702  000040          LFGLP1: MOVB    #40,R2                  ;INITIALIZE ASCII CHAR
 9434  036304' 110224                  LFGLP2: MOVB    R2,(R4)+                ;MOVE THE CHAR
 9435  036306' 005303                          DEC     R3                      ;DECREMENT BYTE COUNT
 9436  036310' 001405                          BEQ     LFGEXT                  ;EXIT IF ZERO
 9437  036312' 105202                          INCB    R2                      ;CREATE NEXT CHAR
 9438  036314' 122702  000140                  CMPB    #140,R2                 ;LAST CHAR OF SET
 9439  036320' 001371                          BNE     LFGLP2                  ;NO-GO MOVE IT
 9440  036322' 000766                          BR      LFGLP1                  ;YES-GO INITIALIZE IT
 9441  036324' 000205                  LFGEXT: RTS     R5                      ;RETURN
 9442                                  ;
 9443                                  ;       FILL WITH PATTERN SUBROUTINE
 9444                                  ;
 9445  036326' 012504                  LFILP:  MOV     (R5)+,R4                ;GET ADDRESS AND
 9446  036330' 012503                          MOV     (R5)+,R3                ;BYTE COUNT
 9447  036332' 012501                          MOV     (R5)+,R1
 9448  036334' 010702                          MOV     PC,R2                   ;DETERMINE PATTERN
 9449  036336' 061102                  LPBASE: ADD     (R1),R2                 ;GENERATION ROUTINE
 9450  036340' 010207                          MOV     R2,PC                   ;THEN GO THERE
 9451
 9452  036342' 012700  000020          LPAT0:  MOV     #16.,R0                 ;WALKING ONES
 9453  036346' 012702  100000                  MOV     #100000,R2
 9454  036352' 004767  000132          10$:    JSR     PC,STRPAT
 9455  036356' 005703                          TST     R3
 9456  036360' 001452                          BEQ     FILPEX
 9457  036362' 000241                          CLC
 9458  036364' 006002                          ROR     R2
 9459  036366' 005300                          DEC     R0
 9460  036370' 001370                          BNE     10$
 9461  036372' 000763                          BR      LPAT0
 9462  036374' 012700  000020          LPAT1:  MOV     #16.,R0                 ;WALKING ZEROES
 9463  036400' 012702  077777                  MOV     #77777,R2
 9464  036404' 004767  000100          10$:    JSR     PC,STRPAT
 9465  036410' 005703                          TST     R3
 9466  036412' 001435                          BEQ     FILPEX
 9467  036414' 000261                          SEC
 9468  036416' 006002                          ROR     R2
 9469  036420' 005300                          DEC     R0
 9470  036422' 001370                          BNE     10$
 9471  036424' 000763                          BR      LPAT1
 9472  036426' 012700  000020          LPAT2:  MOV     #16.,R0                 ;EXPANDING ONES
 9473  036432' 012702  100000                  MOV     #100000,R2
 9474  036436' 004767  000046          10$:    JSR     PC,STRPAT
 9475  036442' 005703                          TST     R3
 9476  036444' 001420                          BEQ     FILPEX
 9477  036446' 006202                          ASR     R2
 9478  036450' 005300                          DEC     R0
 9479  036452' 001371                          BNE     10$
 9480  036454' 000764                          BR      LPAT2
 9481  036456' 012700  000020          LPAT3:  MOV     #16.,R0                 ;EXPANDING ZEROES
 9482  036462' 012702  077777                  MOV     #77777,R2
 9483  036466' 004767  000016          10$:    JSR     PC,STRPAT
 9484  036472' 005703                          TST     R3
 9485  036474' 001404                          BEQ     FILPEX
 9486  036476' 006202                          ASR     R2
 9487  036500' 005300                          DEC     R0
 9488  036502' 001371                          BNE     10$
```

```
 9489  036504' 000764                    BR      LPAT3
 9490  036506' 000205          FILPEX:   RTS     R5              ;RETURN INLINE
 9491  036510' 110224          STRPAT:   MOVB    R2,(R4)+        ;STORE PATTERN SUBR
 9492  036512' 005303                    DEC     R3
 9493  036514' 001404                    BEQ     STPTEX
 9494  036516' 000302                    SWAB    R2
 9495  036520' 110224                    MOVB    R2,(R4)+
 9496  036522' 005303                    DEC     R3
 9497  036524' 000302                    SWAB    R2
 9498  036526' 000207          STPTEX:   RTS     PC
 9499  036530' 012700 125252    LPAT4:   MOV     #125252,R0      ;ALTERNATE ONES (HORIZ)
 9500  036534' 012701 125252             MOV     #125252,R1
 9501  036540' 000450                    BR      LPATCM
 9502  036542' 012700 052525    LPAT5:   MOV     #052525,R0      ;ALTERNATE ZEROES (HORIZ)
 9503  036546' 012701 052525             MOV     #052525,R1
 9504  036552' 000443                    BR      LPATCM
 9505  036554' 012700 177777    LPAT6:   MOV     #177777,R0      ;ALTERNATE ONES (VERT)
 9506  036560' 012701 000000             MOV     #0,R1
 9507  036564' 000436                    BR      LPATCM
 9508  036566' 012700 000000    LPAT7:   MOV     #0,R0           ;ALTERNATE ZEROES (VERT)
 9509  036572' 012701 177777             MOV     #177777,R1
 9510  036576' 000431                    BR      LPATCM
 9511  036600' 012700 070707    LPAT8:   MOV     #70707,R0       ;OCTAL CHECKERBOARD
 9512  036604' 012701 107070             MOV     #107070,R1
 9513  036610' 000424                    BR      LPATCM
 9514  036612' 012700 125252    LPAT9:   MOV     #125252,R0      ;BINARY CHECKERBOARD
 9515  036616' 012701 052525             MOV     #52525,R1
 9516  036622' 000417                    BR      LPATCM
 9517  036624' 005002           LPATA:   CLR     R2              ;START WITH WORD OF ZEROES
 9518  036626' 110224           10$:     MOVB    R2,(R4)+        ;STORE A BYTE
 9519  036630' 005303                    DEC     R3
 9520  036632' 001415                    BEQ     LPATEX          ;EXIT IF DONE
 9521  036634' 000302                    SWAB    R2              ;SWAP BYTES WITHIN R2
 9522  036636' 110224                    MOVB    R2,(R4)+        ;STORE A BYTE
 9523  036640' 000302                    SWAB    R2              ;SWAP BYTES WITHIN R2
 9524  036642' 005303                    DEC     R3
 9525  036644' 001410                    BEQ     LPATEX          ;EXIT IF DONE
 9526  036646' 005202                    INC     R2              ;ADD 1 TO R2
 9527  036650' 000766                    BR      10$             ;GO AGAIN
 9528  036652' 012700 165555    LPATB:   MOV     #165555,R0      ;DISK WORST CASE
 9529  036656' 012701 133333             MOV     #133333,R1
 9530  036662' 004767 000002    LPATCM:  JSR     PC,PATGEN
 9531  036666' 000205           LPATEX:  RTS     R5
 9532  036670' 010002           PATGEN:  MOV     R0,R2           ;GET WORD FROM R0
 9533  036672' 110224                    MOVB    R2,(R4)+        ;STORE A BYTE
 9534  036674' 005303                    DEC     R3
 9535  036676' 001414                    BEQ     PGENEX          ;EXIT IF DONE
 9536  036700' 000302                    SWAB    R2              ;SWAP BYTES WITHIN R2
 9537  036702' 110224                    MOVB    R2,(R4)+        ;STORE A BYTE
 9538  036704' 005303                    DEC     R3              ;DONE ?
 9539  036706' 001410                    BEQ     PGENEX          ;YES - EXET
 9540  036710' 010102                    MOV     R1,R2           ;GET WORD FROM R1
 9541  036712' 110224                    MOVB    R2,(R4)+        ;STORE A BYTE
 9542  036714' 005303                    DEC     R3              ;DONE ?
 9543  036716' 001404                    BEQ     PGENEX          ;YES - EXIT
 9544  036720' 000302                    SWAB    R2              ;SWAP BYTES WITHIN R2
```

```
 9545  036722' 110224                          MOVB    R2,(R4)+                ;STORE A BYTE
 9546  036724' 005303                          DEC     R3                      ;DONE ?
 9547  036726' 001360                          BNE     PATGEN                  ;NO - GET NEXT WORD
 9548  036730' 000207                  PGENEX: RTS     PC                      ;EXIT
 9549
 9550                                  ;
 9551                                  ;       ROTATE STRING SUBROUTINE
 9552                                  ;
 9553  036732' 012504                  LROTR:  MOV     (R5)+,R4                ;GET STRING ADDRESS
 9554  036734' 012503                          MOV     (R5)+,R3                ;GET # BYTES TO BE SPUN
 9555  036736' 010402                          MOV     R4,R2
 9556  036740' 005303                          DEC     R3
 9557  036742' 060302                          ADD     R3,R2                   ;SET UP RIGHT POINTER
 9558  036744' 010201                          MOV     R2,R1                   ;SET UP LEFT POINTER
 9559  036746' 111200                          MOVB    (R2),R0                 ;SAVE RIGHT HAND BYTE
 9560  036750' 005202                          INC     R2
 9561  036752' 114142                  LROTRL: MOVB    -(R1),-(R2)             ;MOVE OTHER BYTES RIGHT 1
 9562  036754' 020204                          CMP     R2,R4
 9563  036756' 001375                          BNE     LROTRL
 9564  036760' 110012                          MOVB    R0,(R2)                 ;MOVE SAVED BYTE IN ON LEFT
 9565  036762' 000205                          RTS     R5                      ;RETURN
 9566                                  ;
 9567                                  ;       DELAY SUBROUTINE
 9568                                  ;
 9569  036764' 012500                  LDLAY:  MOV     (R5)+,R0                ;GET DELAY IN MILLISECONDS
 9570  036766' 012701  000522          LDLLP1: MOV     #522,R1                 ;LOAD CONSTANT FOR 1 MS
 9571  036772' 005301                  LDLLP2: DEC     R1
 9572  036774' 001376                          BNE     LDLLP2                  ;DELAY 1 MILLISECOND
 9573  036776' 005300                          DEC     R0
 9574  037000' 001372                          BNE     LDLLP1                  ;REPEAT AS PER COUNT
 9575  037002' 000205                          RTS     R5                      ;RETURN
 9576                                  ;
 9577                                  ;       VERIFY SUBROUTINE
 9578                                  ;
 9579  037004' 012500                  LVERFY: MOV     (R5)+,R0                ;GET BYTE COUNT
 9580  037006' 012501                          MOV     (R5)+,R1                ;GET 1ST DATA ADR
 9581  037010' 012502                          MOV     (R5)+,R2                ;GET 2ND DATA ADR
 9582  037012' 005004                          CLR     R4                      ;INITIALIZE BYTE #
 9583  037014' 004767  000254                  JSR     PC,LVECMP               ;DO DATA COMPARE
 9584  037020' 000401                          BR      10$                     ;MISCOMPARE? (N,Y-10$)
 9585  037022' 000205                          RTS     R5                      ;EXIT IN-LINE TO USER PROG
 9586
 9587                                          ;VERIFY MISCOMPARE
 9588
 9589              001                         .IF DF MM
 9590                                          .IFF
 9591  037024' 016703  175636          10$:    MOV     CPGCPT,R3               ;GET ADR OF PROG'S TABLE
 9592                                          .IFT
 9593                                  10$:    CLR     R3                      ;GET ADR OF PROG'S TABLE
 9594              000                         .ENDC
 9595  037030' 010301                          MOV     R3,R1                   ;POINT AT ADR OF
 9596  037032' 062701  000304                  ADD     #PTLGTH+DECTAD,R1       ;DATA ERROR CNTR
 9597  037036' 012761  000001  177760          MOV     #1,DEVIW8-DECTAD(R1)    ;SET THE ERROR INDICATOR
 9598  037044' 005711                          TST     (R1)                    ;IS THERE AN ERROR CNTR?
 9599  037046' 001402                          BEQ     20$                     ;Y,N-20$
 9600  037050' 061101                          ADD     (R1),R1                 ;MAKE ITS ADR ABSOLUTE
```

```
9601  037052' 005211                          INC     (R1)                    ;ADD 1 TO PROG'S DATA ERR CNT
9602  037054' 032763  020000  000002  20$:    BIT     #PRONER,POPSW(R3)       ;OPSW INDICATE PRINT ON ERROR?
9603  037062' 001403                          BEQ     30$                     ;N,Y-30$
9604  037064' 062706  000012                  ADD     #10.,SP                 ;REMOVE ERROR DATA FROM STACK
9605  037070' 000205                          RTS     R5                      ;EXIT IN-LINE TO USER PROG
9606  037072' 004567  001520          30$:    JSR     R5,LTSTDE               ;GO TEST DEDICATED FLAGS
9607  037076' 105067  000571                  CLRB    LPRRDE                  ;RESET REALLY DEDICATED FLAG
9608  037102' 016300  000022                  MOV     PSRCST(R3),R0           ;GET START ADR OF SRC STMNT'S
9609  037106' 010501                          MOV     R5,R1                   ;COMPUTE THIS STMNT'S ADR
9610  037110' 162701  000012                  SUB     #10.,R1
9611  037114' 026001  000004          45$:    CMP     4(R0),R1                ;THIS THE STMNT?
9612  037120' 001403                          BEQ     50$                     ;N,Y-50$
9613  037122' 111002                          MOVB    (R0),R2                 ;GET THIS STMNT'S LENGTH
9614  037124' 060200                          ADD     R2,R0                   ;POINT AT NXT STMNT
9615  037126' 000772                          BR      45$                     ;GO CK IT
9616  037130' 005720                  50$:    TST     (R0)+                   ;POINT AT STMNT #
9617  037132' 010701                          MOV     PC,R1                   ;SET UP ADR OF AREA IN MSG
9618  037134' 062701  000262                  ADD     #LVEHDN-.,R1
9619  037140' 004567  001660                  JSR     R5,DECASC               ;CONVERT STMNT # TO ASCII
9620  037144' 012767  006451  000250          MOV     #06451,LVEHDN+4         ;RESTORE ) & CR DATA IN THE MSG
9621  037152' 004567  000600                  JSR     R5,LPROGN               ;PRINT THE PROG #
9622  037156' 004567  000326                  JSR     R5,USRPRT               ;PRINT THE HEADER MSG
9623  037162' 037372'                 LVECL1: .WORD   LVEHDR
9624  037164' 000064                          .WORD   52.
9625  037166' 010400                  LVEDEP: MOV     R4,R0                   ;GET THE BAD BYTE # IN R0
9626  037170' 004567  001770                  JSR     R5,BTASLB               ;CONVERT TO ASCII DECIMAL
9627  037174' 000262                          .WORD   LVEMSG-.
9628  037176' 005201                          INC     R1                      ;POINT AT MSG'S 1ST DATA FIELD
9629  037200' 012600                          MOV     (SP)+,R0                ;GET FIRST BAD DATA BYTE
9630  037202' 004767  000136                  JSR     PC,LVECDB               ;CONVERT IT TO ASCII
9631  037206' 012600                          MOV     (SP)+,R0                ;GET 2ND BAD DATA BYTE
9632  037210' 005721                          TST     (R1)+                   ;POINT AT MSG'S 2ND DATA FIELD
9633  037212' 004767  000126                  JSR     PC,LVECDB               ;CONVERT IT TO ASCII
9634  037216' 004567  000534                  JSR     R5,LPROGN               ;PRINT THE PROG # MSG
9635  037222' 004567  000262                  JSR     R5,USRPRT               ;PRINT THE BAD DATA MSG
9636  037226' 037456'                 LVECL2: .WORD   LVEMSG
9637  037230' 000032                          .WORD   26.
9638  037232' 012600                          MOV     (SP)+,R0                ;RESTORE THE CURR BYTE COUNT
9639  037234' 012601                          MOV     (SP)+,R1                ;RESTORE DATA FIELD # 1 ADR
9640  037236' 012602                          MOV     (SP)+,R2                ;RESTORE DATA FIELD # 2 ADR
9641  037240' 032763  000002  000002          BIT     #PFBBOV,POPSW(R3)       ;OPSW SAY PRINT ONLY 1ST BAD BYTE?
9642  037246' 001003                          BNE     LVEREX                  ;N,Y-LVEREX
9643  037250' 004767  000020                  JSR     PC,LVECMP               ;RESUME DATA COMPARE
9644  037254' 000744                          BR      LVEDEP                  ;ANOTHER ERROR? (N,Y-LVEDEP)
9645  037256' 042767  020000  174730  LVEREX: BIC     #CTUDED,CSYSFW          ;RESET SYSTEM'S DEDICATED FLAG
9646  037264' 042713  000040                  BIC     #SETDED,(R3)            ;RESET PROG'S DEDICATED FLAG
9647  037270' 000167  175464                  JMP     CUPGER                  ;GO TO MPG'S USER PROG ERR RETN POINT
9648
9649
9650  037274' 005700                  LVECMP: TST     R0                      ;BYTE COUNT = 0?
9651  037276' 001003                          BNE     100$                    ;Y,N-100$
9652  037300' 062716  000002                  ADD     #2,(SP)                 ;POINT PAST ERROR BR INST
9653  037304' 000207                          RTS     PC                      ;EXIT IN-LINE AFTER BR INST
9654  037306' 005204                  100$:   INC     R4                      ;ADD 1 TO BYTE #
9655  037310' 005300                          DEC     R0                      ;SUB 1 FROM BYTE CNT
9656  037312' 122122                          CMPB    (R1)+,(R2)+             ;DATA BYTES COMPARE?
```

```
 9657  037314' 001767                        BEQ     LVECMP                  ;N,Y-LVECMP
 9658  037316' 012627                        MOV     (SP)+,(PC)+             ;SAVE RETURN ADR
 9659  037320' 000000              110$:     .WORD   0
 9660  037322' 010246                        MOV     R2,-(SP)                ;SAVE CURR DATA 2 ADR
 9661  037324' 010146                        MOV     R1,-(SP)                ;SAVE CURR DATA 1 ADR
 9662  037326' 010046                        MOV     R0,-(SP)                ;SAVE CURR BYTE CNT
 9663  037330' 116246  177777                MOVB    -1(R2),-(SP)            ;SAVE BAD DATA # 2 BYTE
 9664  037334' 116146  177777                MOVB    -1(R1),-(SP)            ;SAVE BAD DATA # 1 BYTE
 9665  037340' 016707  177754                MOV     110$,PC                 ;EXIT IN-LINE & DO THE BR INST
 9666
 9667
 9668  037344' 042700  177400      LVECDB:   BIC     #177400,R0              ;RESET OTHER BYTE
 9669  037350' 004567  001552                JSR     R5,BINAS1               ;CONVERT BYTE TO ASCII
 9670  037354' 012761  020040  177772        MOV     #20040,-6(R1)           ;STORE LEADING BLANKS
 9671  037362' 112761  000040  177774        MOVB    #40,-4(R1)
 9672  037370' 000207                        RTS     PC                      ;EXIT IN-LINE
 9673
 9674
 9675  037372' 040504  040524  042440  LVEHDR:  .ASCII  /DATA ERROR (STMNT # /
       037400' 051122  051117  024040
       037406' 052123  047115  020124
       037414' 020043
 9676  037416' 054130  054130  006451  LVEHDN:  .ASCII  /XXXX)/<015><012><011>
       037424' 004412
 9677  037426' 054502  042524  021440          .ASCII  /BYTE #  DATA 1  DATA 2/<015><012>
       037434' 020040  040504  040524
       037442' 030440  020040  040504
       037450' 040524  031040  005015
 9678                                          .EVEN
 9679  037456' 020040  020040  020040  LVEMSG:  .ASCII  /                       /<015><012>
       037464' 020040  020040  020040
       037472' 020040  020040  020040
       037500' 020040  020040  020040
       037506' 005015
 9680                                          .EVEN
```

```
                                .SBTTL  USER PRINT SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE HANDLES ALL PRINT OPERATIONS TO
                        ;       THE DISPLAY DEVICE AS A RESULT OF
                        ;       USER INSTRUCTIONS.  ITS PURPOSE IS TO DEDICATE
                        ;       THE PRINT DEVICE TO ONE USER PROGRAM UNTIL
                        ;       A MULTIPLE LINE OUTPUT IS PRINTED WHILE
                        ;       ALLOWING INTERACTION AMONG OTHER I/O DEVICES
                        ;       BETWEEN LINES.
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,USRPRT
                        ;                               .WORD   ADDRESS
                        ;                               .WORD   BYTE COUNT
                        ;
                        ;*****************************************************************

037510' 105067  000156          USRPRT: CLRB    LLPRTF                  ;RESET LAST PRINT FLAG
037514' 004467  001256          USRPR1: JSR     R4,SREG                 ;SAVE REG'S R0 - R4
037520' 012504                          MOV     (R5)+,R4                ;GET TEXT ADDRESS
037522' 012503                          MOV     (R5)+,R3                ;AND BYTE COUNT
037524' 010301                          MOV     R3,R1                   ;SET UP WORK COUNT
037526' 005303                          DEC     R3                      ;ADJ TOTAL CNT FOR LATER TESTING
037530' 020127  000062          USRLP:  CMP     R1,#50.                 ;WILL MSG FIT IN BUF?
037534' 003402                          BLE     10$                     ;YES - GO AHEAD
037536' 012701  000062                  MOV     #50.,R1                 ;NO-TRUNCATE IT FOR NOW
037542' 010167  000064          10$:    MOV     R1,LPRCNT               ;STORE BYTE CNT
037546' 010702                          MOV     PC,R2                   ;POINT R2 AT
037550' 062702  000124                  ADD     #LPRBUF-.,R2            ;OUTPUT BUFFER
037554' 112422                  20$:    MOVB    (R4)+,(R2)+             ;MOVE MESSAGE
037556' 005303                          DEC     R3                      ;DECR TOTAL CNT
037560' 005301                          DEC     R1                      ;DECR WORK CNT
037562' 001374                          BNE     20$                     ;GOT ALL CHARS THIS GROUP? (Y,N-20$)
037564' 005703                          TST     R3                      ;MORE DATA AFTER THIS GROUP?
037566' 100006                          BPL     LPRCAL                  ;N,Y-LPRCAL
037570' 105767  000076                  TSTB    LLPRTF                  ;THIS THE LAST PRINT?
037574' 001403                          BEQ     LPRCAL                  ;Y,N-LPRCAL
037576' 052767  040000  174410          BIS     #CRUDED,CSYSFW          ;TELL DRIVER TO RESET DEDICATED FLAG
                    001                 .IF DF MM
                                        .IFF
037604' 016701  140226          LPRCAL: MOV     CTUSWR,R1               ;INITIALIZE TO CONS TERM USR WRITE
037610' 016700  140200                  MOV     CLISTI,R0               ;GET ADR OF LIST DEV
037614' 005760  000004                  TST     4(R0)                   ;IS THERE A PRINTER ASSIGNED?
037620' 001402                          BEQ     30$                     ;Y,N-30$
037622' 016701  140214                  MOV     CUSPRT,R1               ;GET USER'S PRINT ROUT ADR
037626' 004511                  30$:    JSR     R5,(R1)                 ;CALL FOR PRINT
037630' 000044                          .WORD   LPRBUF-.                ;ADDRESS
037632' 000000                  LPRCNT: .WORD   XXXX                    ;BYTE COUNT
037634' 000030                          .WORD   LPRBSY-.                ;BUSY ADDRESS
037636' 000000                  LPRFWA: .WORD   0                       ;FLAGWORD ADDRESS
                                        .IFT
                                LPRCAL: TRAP    10                      ;GO INTO KERNEL MODE. DOES
                                                                        ;INSTS AT "UPRTKM" & RETURNS
                                                                        ;IN-LINE IF NOT BUSY. IF BUSY,
```

```
                                                                        ;RETURNS AT "LPRBSY".
                    000                         .ENDC
037640' 004567  175352                          JSR     R5,CIOBSY       ;WAIT FOR I/O COMPLETE
037644' 005703                                  TST     R3              ;MORE DATA ON THIS MSG?
037646' 100403                                  BMI     USRPEX          ;BR IF NO - GO TO EXIT
037650' 010301                                  MOV     R3,R1           ;SET UP REMAINING BYTE CNT
037652' 005201                                  INC     R1
037654' 000725                                  BR      USRLP           ;GO PRINT NEXT GROUP
037656' 004467  001126          USRPEX:         JSR     R4,RREG         ;RELOAD REGISTERS
037662' 000205                                  RTS     R5              ;RETURN
037664' 004567  175326          LPRBSY:         JSR     R5,CIOBSY       ;IF I/O NOT STARTED, WAIT & THEN
037670' 000745                                  BR      LPRCAL          ;HIT IT AGAIN
                    001                         .IF DF MM
                                UPRTKM:         MOV     CTUSWR,R1       ;INITIALIZE TO CONS TERM USR WRITE
                                                MOV     CLISTI,R0       ;GET ADR OF LIST DEV
                                                TST     4(R0)           ;IS THERE A PRINTER ASSIGNED?
                                                BEQ     30$             ;Y,N-30$
                                                MOV     CUSPRT,R1       ;GET USER'S PRINT ROUT ADR
                                30$:            JSR     R5,(R1)         ;ISSUE THE MSG
                                                .WORD   LPRBUF-.        ;  REL DATA ADR
                                LPRCNT:         .WORD   XXXX            ;  MSG BYTE CNT
                                                .WORD   UPRTKB-.        ;  REL BUSY ADR
                                LPRFWA:         .WORD   XXXX            ;  FLAGWORD ADR
                                                RTI                     ;RETURN IN-LINE FROM TRAP

                                UPRTKB:         MOV     #LPRBSY-PG6BA+P6CONS,(SP) ;SET UP BUSY RETURN ADR
                                                RTI                     ;RETURN TO BUSY ADR FROM TRAP
                    000                         .ENDC
037672'     000                 LLPRTF:         .BYTE   0
037673'     000                 LPRRDE:         .BYTE   0               ;REALLY DEDICATED FLAG
037674' 000062                  LPRBUF:         .BLKB   50.             ;MESSAGE BUFFER
                                ;
                                ;
                                ;       PRINT PROGRAM NUMBER SUBROUTINE
                                ;
037756' 105767  177711          LPROGN:         TSTB    LPRRDE          ;TEST REALLY DED FLAG
037762' 001013                                  BNE     LPNEX           ;DONT PRINT P NBR IF SET
037764' 010246                                  MOV     R2,-(SP)        ;SAVE R2
                    001                         .IF DF MM
                                                .IFF
037766' 016702  177644                          MOV     LPRFWA,R2       ;GET PROG TBL POINTER
                                                .IFT
                                                CLR     R2              ;POINT AT START OF PROG TBL
                    000                         .ENDC
037772' 016267  000006  000016                  MOV     PASCIN(R2),LPN+2 ;MOVE P NBR TO MSG
040000' 012602                                  MOV     (SP)+,R2        ;RESTORE R2
040002' 004567  177502                          JSR     R5,USRPRT       ;PRINT  P NN
040006' 040014'                 LPNADR:         .WORD   LPN
040010' 000006                                  .WORD   6
040012' 000205                  LPNEX:          RTS     R5              ;RETURN
040014' 021520                  LPN:            .ASCII  /P#/
040016'     040     040     040                 .BYTE   40,40,40,11
040021'     011
                                ;
                                ;       PRINT ASCII SUBROUTINE
                                ;
```

```
 9793  040022' 004567 000570          LPRNG:  JSR     R5,LTSTDE               ;GO TEST DEDICATE FLAG
 9794  040026' 004567 177724          UPRASC: JSR     R5,LPROGN               ;PRINT  P NN
 9795  040032' 012500                          MOV     (R5)+,R0                ;GET ADDRESS
 9796  040034' 012501                          MOV     (R5)+,R1                ;AND BYTE COUNT
 9797  040036' 100422                          BMI     LPGNCT                  ;BRANCH IF NEG COUNT
 9798  040040' 105067 177627                   CLRB    LPRRDE
 9799  040044' 020500                  LPGTIL: CMP     R5,R0                   ;TEXT INLINE?
 9800  040046' 001005                          BNE     LPGTG1                  ;NO-CONTINUE
 9801  040050' 060105                          ADD     R1,R5                   ;YES-ADJUST RETURN
 9802  040052' 032705 000001                   BIT     #1,R5                   ;WAS BYTE CNT ODD?
 9803  040056' 001401                          BEQ     LPGTG1                  ;Y,N-LPGTG1
 9804  040060' 005205                          INC     R5                      ;ADJ RET ADR TO EVEN
 9805  040062' 010067 000010          LPGTG1: MOV     R0,LPGADR               ;MOVE ADDRESS
 9806  040066' 010167 000006                   MOV     R1,LPGBYC               ;AND BYTE COUNT TO CALL
 9807  040072' 004567 177412                   JSR     R5,USRPRT               ;CALL USER PRINT SUB
 9808  040076' 000000                  LPGADR: .WORD   0
 9809  040100' 000000                  LPGBYC: .WORD   0
 9810  040102' 000475                          BR      LPOFXT                  ;CR LF AND EXIT
 9811  040104' 105267 177563          LPGNCT: INCB    LPRRDE                  ;SET REALLY DEDICATED FL
 9812  040110' 005401                          NEG     R1                      ;GET POSITIVE COUNT
 9813  040112' 000754                          BR      LPGTIL                  ;GO TEST IF TEXT INLINE
 9814                                  ;
 9815                                  ;
 9816                                  ;       PRINT OCTAL OR DECIMAL SUBROUTINE
 9817                                  ;
 9818  040114' 004567 000476          LPRND:  JSR     R5,LTSTDE               ;GO TEST DEDICATE FLAG
 9819  040120' 005067 000254                   CLR     LOCTSW                  ;CLEAR OCTAL SWITCH
 9820  040124' 000404                          BR      LPOSDF
 9821  040126' 004567 000464          LPRNO:  JSR     R5,LTSTDE               ;GO TEST DEDICATE FLAG
 9822  040132' 005267 000242                   INC     LOCTSW                  ;SET OCTAL SWITCH
 9823  040136' 012767 000010 000160  LPOSDF: MOV     #8.,LPOCNT              ;SET FORMAT CNTR =  8
 9824  040144' 004567 177606                   JSR     R5,LPROGN               ;PRINT  P NN
 9825  040150' 105067 177517                   CLRB    LPRRDE                  ;CLEAR REALLY DED FLAG
 9826  040154' 012504                          MOV     (R5)+,R4                ;GET ADDRESS
 9827  040156' 012503                          MOV     (R5)+,R3                ;GET BYTE COUNT
 9828  040160' 032704 000001          LPOOET: BIT     #1,R4                   ;IS ADDRESS ODD?
 9829  040164' 001431                          BEQ     LPOCBC                  ;NO-CHECK BYTE COUNT
 9830  040166' 112400                  LPOODD: MOVB    (R4)+,R0                ;GET ONE BYTE
 9831  040170' 042700 177400                   BIC     #177400,R0
 9832  040174' 004567 000152                   JSR     R5,LTOASC               ;CONVERT
 9833  040200' 010702                          MOV     PC,R2
 9834  040202' 062702 174060                   ADD     #LINBUF-.,R2
 9835  040206' 012722 020040                   MOV     #20040,(R2)+            ;BLANK OUT
 9836  040212' 112722 000040                   MOVB    #40,(R2)+               ;LEADING ZEROES
 9837  040216' 062702 000003                   ADD     #3,R2
 9838  040222' 012722 020040          LPOPRT: MOV     #20040,(R2)+            ;STORE TWO TRAILING BLANKS
 9839  040226' 004567 177256                   JSR     R5,USRPRT               ;CALL USER PRINT SUB
 9840  040232' 034262'                 LPOADR: .WORD   LINBUF
 9841  040234' 000010                          .WORD   8.
 9842  040236' 005303                          DEC     R3                      ;DECREMENT BYTE COUNT
 9843  040240' 001416                          BEQ     LPOFXT                  ;IF ZERO FIND EXIT
 9844  040242' 005367 000056                   DEC     LPOCNT                  ;DECREMENT FORMAT COUNTER
 9845  040246' 001427                          BEQ     LPORTN                  ;CR - LF IF ZERO
 9846  040250' 022703 000001          LPOCBC: CMP     #1,R3                   ;IF ONLY ONE REMAINING
 9847  040254' 001744                          BEQ     LPOODD                  ;TREAT AS ODD ADDRESS
 9848  040256' 012400                  LPOEVN: MOV     (R4)+,R0                ;GET WORD
```

```
 9849  040260' 004567  000066                    JSR     R5,LTOASC               ;CONVERT
 9850  040264' 010702                            MOV     PC,R2
 9851  040266' 062702  174002                    ADD     #LINBUF+6-.,R2
 9852  040272' 005303                            DEC     R3                      ;DECREMENT BYTE CNT BY 1
 9853  040274' 000752                            BR      LPOPRT                  ;GO PRINT IT
 9854  040276' 105767  177371          LPOFXT:   TSTB    LPRRDE                  ;TEST REALLY DEDICATED FL
 9855  040302' 001007                            BNE     LPONOM                  ;IF SET THERE'S MORE TO GO
 9856  040304' 112767  000001  177360  ULCRLF:   MOVB    #1,LLPRTF               ;SET THIS IS THE LAST PRT FLG
 9857  040312' 004567  177176          UPCRLF:   JSR     R5,USRPR1               ;PRINT A CARRIAGE RET
 9858  040316' 042114'                  LPOCR1:   .WORD   LCRETN                  ;AND LINE FEED
 9859  040320' 000002                            .WORD   2
 9860  040322' 000205                  LPONOM:   RTS     R5                      ;RETURN
 9861  040324' 000000                  LPOCNT:   .WORD   0                       ;FORMAT COUNTER
 9862  040326' 004567  177156          LPORTN:   JSR     R5,USRPRT               ;PRINT CR LF
 9863  040332' 042114'                  LPOCR2:   .WORD   LCRETN
 9864  040334' 000002                            .WORD   2
 9865  040336' 012767  000010  177760            MOV     #8.,LPOCNT              ;RE-INIT COUNTER
 9866  040344' 004567  177406                    JSR     R5,LPROGN               ;PRINT  P NN
 9867  040350' 000737                            BR      LPOCBC                  ;GO CHECK BYTE COUNT
 9868  040352' 005767  000022          LTOASC:   TST     LOCTSW                  ;TEST OCTAL SWITCH
 9869  040356' 001404                            BEQ     LTOD                    ;BRANCH IF NOT  SET
 9870  040360' 004567  000536          LTOO:     JSR     R5,BINASC               ;CONVERT TO OCTAL
 9871  040364' 173676                            .WORD   LINBUF-.
 9872  040366' 000205                            RTS     R5                      ;RETURN
 9873  040370' 004567  000570          LTOD:     JSR     R5,BTASLB               ;CONVERT TO DECIMAL
 9874  040374' 173666                            .WORD   LINBUF-.
 9875  040376' 000205                            RTS     R5                      ;RETURN
 9876  040400' 000000                  LOCTSW:   .WORD   0
 9877                                    ;
 9878                                    ;
 9879                                    ;       PRINT BINARY SUBROUTINE
 9880                                    ;
 9881  040402' 004567  000210          LPRNB:    JSR     R5,LTSTDE               ;GO TEST DEDICATE FLAG
 9882  040406' 012767  000002  177710            MOV     #2,LPOCNT               ;SET FORMAT CTR=2
 9883  040414' 004567  177336                    JSR     R5,LPROGN               ;PRINT  P NN
 9884  040420' 105067  177247                    CLRB    LPRRDE                  ;CLEAR REALLY DED FLAG
 9885  040424' 012504                            MOV     (R5)+,R4                ;GET ADDRESS
 9886  040426' 012503                            MOV     (R5)+,R3                ;GET BYTE COUNT
 9887  040430' 032704  000001          LPBOET:   BIT     #1,R4                   ;IS ADDRESS ODD?
 9888  040434' 001443                            BEQ     LPBCBC                  ;NO-CHECK BYTE COUNT
 9889  040436' 010702                  LPBODD:   MOV     PC,R2
 9890  040440' 062702  173622                    ADD     #LINBUF-.,R2
 9891  040444' 012701  000010                    MOV     #8.,R1                  ;SET SPACE COUNT IN R1
 9892  040450' 112722  000040          LPBLP1:   MOVB    #40,(R2)+               ;INSERT
 9893  040454' 005301                            DEC     R1                      ;8 LEADING BLANKS
 9894  040456' 001374                            BNE     LPBLP1
 9895  040460' 012701  000010                    MOV     #8.,R1                  ;SET BIT COUNT=8
 9896  040464' 112400                            MOVB    (R4)+,R0                ;GET A BYTE
 9897  040466' 000300                            SWAB    R0                      ;LEFT JUSTIFY
 9898  040470' 006300                  LPBLP2:   ASL     R0                      ;SHIFT LEFT INTO CARRY
 9899  040472' 103403                            BCS     LPBTG1                  ;IF CARRY NOT SET
 9900  040474' 112722  000060                    MOVB    #60,(R2)+               ;STORE ASCII ZERO
 9901  040500' 000402                            BR      LPBTG2
 9902  040502' 112722  000061          LPBTG1:   MOVB    #61,(R2)+               ;OTHERWISE ASCII ONE
 9903  040506' 005301                  LPBTG2:   DEC     R1                      ;REPEAT UNTIL
 9904  040510' 001367                            BNE     LPBLP2                  ;BIT COUNTER=0
```

```
 9905  040512' 112722  000011                  MOVB    #11.,(R2)+          ;STORE TWO
 9906  040516' 112722  000011                  MOVB    #11,(R2)+           ;TAB CHARACTERS
 9907  040522' 004567  176762                  JSR     R5,USRPRT           ;PRINT 18 CHARS.
 9908  040526' 034262'                 LPBADR: .WORD   LINBUF
 9909  040530' 000022                          .WORD   18.
 9910  040532' 005303                          DEC     R3                  ;DECREMENT BYTE COUNT
 9911  040534' 001660                          BEQ     LPOFXT              ;IF 0, EXIT
 9912  040536' 005367  177562          LPBCFM: DEC     LPOCNT              ;DECREMENT FORMAT COUNTER
 9913  040542' 001413                          BEQ     LPBRTN              ;IF ZERO PRINT CR-LF
 9914  040544' 022703  000001          LPBCBC: CMP     #1,R3               ;IF ONLY ONE BYTE LEFT
 9915  040550' 001732                          BEQ     LPBODD              ;TREAT AS ODD
 9916  040552' 010702                          MOV     PC,R2
 9917  040554' 062702  173506                  ADD     #LINBUF-.,R2
 9918  040560' 012400                  LPBEVN: MOV     (R4)+,R0            ;GET A WORD IF EVEN
 9919  040562' 012701  000020                  MOV     #16.,R1             ;SET BIT COUNT=16
 9920  040566' 005303                          DEC     R3
 9921  040570' 000737                          BR      LPBLP2              ;GO STORE ZEROES & ONES
 9922  040572' 004567  176712          LPBRTN: JSR     R5,USRPRT           ;PRINT CR-LF
 9923  040576' 042114'                 LPBCR1: .WORD   LCRETN
 9924  040600' 000002                          .WORD   2
 9925  040602' 012767  000002  177514          MOV     #2,LPOCNT           ;RE-INIT FORMAT CTR
 9926  040610' 004567  177142                  JSR     R5,LPROGN           ;PRINT  P NN
 9927  040614' 000753                          BR      LPBCBC              ;GO CHECK BYTE COUNT
 9928
 9929
 9930  040616' 032767  020000  173370  LTSTDE: BIT     #CTUDED,CSYSFW      ;TEST DEDICATE FLAG
 9931  040624' 001410                          BEQ     LNOTDE              ;RETURN IF NOT SET
 9932                  001                     .IF DF MM
 9933                                          .IFF
 9934  040626' 032777  000040  174032          BIT     #SETDED,@CPGCPT     ;CONSOLE DED TO THIS PROG ?
 9935                                          .IFT
 9936                                          BIT     #SETDED,@#0         ;CONSOLE DED TO THIS PROG?
 9937                  000                     .ENDC
 9938  040634' 001004                          BNE     LNOTDE              ;N,Y-LNOTDE
 9939  040636' 162705  000004                  SUB     #4,R5               ;OTHERWISE BACK UP R5
 9940  040642' 000167  174350                  JMP     CIOBSY              ;AND RELEASE CONTROL
 9941                  001                     .IF DF MM
 9942                                          .IFF
 9943  040646' 016703  174014          LNOTDE: MOV     CPGCPT,R3           ;GET PROG'S TABLE ADR
 9944                                          .IFT
 9945                                  LNOTDE: CLR     R3                  ;GET VIRTUAL ADR OF PROG'S TABLE
 9946                  000                     .ENDC
 9947  040652' 052713  000040                  BIS     #SETDED,(R3)        ;SET PROG'S DEDICATED FLAG
 9948  040656' 052767  020000  173330          BIS     #CTUDED,CSYSFW      ;SET SYSTEM'S DEDICATED FLAG
 9949                  001                     .IF DF MM
 9950                                          .IFF
 9951  040664' 010367  176746                  MOV     R3,LPRFWA           ;STORE PROG'S TABLE ADR
 9952                                          .IFT
 9953                                          MOV     PFWADR(R3),LPRFWA   ;STORE HI 16 BITS OF PRG TBL ADR
 9954                  000                     .ENDC
 9955  040670' 000205                          RTS     R5                  ;RETURN
```

```
                                .SBTTL  USER PROGRAM INTERRUPT COMMON PROCESSING ROUTINES

                        ;*****************************************************************
                        ;
                        ;       REQUIRED IN MEMORY MANAGEMENT VERSION
                        ;       INCLUDED IN BOTH VERSIONS
                        ;       FOR DEVICE ROUTINE COMPATIBILITY
                        ;
                        ;*****************************************************************


                        ;SET UP INTERRUPT VECTOR S/R
                        ;
                        ;       JSR     R5,SETVEC               S/R CALL
                        ;       .WORD   VECT ADR                INT VECT ABS ADR
                        ;       .WORD   PSW                     PSW VALUE
                        ;       .WORD   INTROUT-.               INT ROUT REL ADR

            001                 .IF DF MM
                                .IFF
040672' 004467  000100  SETVEC: JSR     R4,SREG                 ;SAVE REGS R0 - R4
040676' 012500                  MOV     (R5)+,R0                ;GET VECTOR ADR
040700' 012501                  MOV     (R5)+,R1                ;GET PSW
040702' 010502                  MOV     R5,R2                   ;GET INT ROUT ADR
040704' 062502                  ADD     (R5)+,R2                ;MAKE IT ABSOLUTE
040706' 010220                  MOV     R2,(R0)+                ;PUT IT IN VECTOR ADR
040710' 010110                  MOV     R1,(R0)                 ;STORE PSW
040712' 000421                  BR      VECEX                   ;GO TO EXIT
                                .IFT
                        SETVEC: JSR     R4,SREG                 ;SAVE  REGISTERS
                                MOV     (R5)+,R1                ;GET VECTOR ADDRESS
                                MOV     (R5)+,R2                ;GET PSW
                                MOV     R5,R3                   ;GET INT ROUT ADR
                                ADD     (R5)+,R3                ;MAKE IT  REL TO PAGE 0
                                MOV     PC,R0                   ;POINT R0 AT
                                ADD     #VECTBL+4-.,R0          ;INTER-PAGE VECTOR TBL
                                MOV     R1,R4
                                ASL     R4                      ;ADJUST R0 TO ENTRY
                                ADD     R4,R0                   ;FOR THIS VECTOR
                                MOV     @#UPAR0,(R0)+           ;SAVE PAR0 CONTENTS FOR PAGE 5 BASE
                                BIS     #P5CONS,R3              ;SET PAR5 SEL BITS IN INT ROUT ADR
                                MOV     R3,(R0)                 ;SAVE IN TBL
                                SUB     #6,R0                   ;POINT R0 AT JSR IN TBL
                                TRAP    4                       ;GO INTO KERN MODE TO SET UP VECT
                                                                ;DOES INST'S AT "KSETVC" & RET IN-LINE
                                BR      VECEX                   ;GO TO EXIT

                        KSETVC: SUB     #P6CONS,R0              ;CONVERT TBL ADR TO KERNEL SPACE
                                ADD     PC,R0
                                ADD     #P66BA-.,R0
                                MOV     R0,(R1)+                ;SET HARDWARE VECTOR
                                MOV     R2,(R1)                 ;AND PSW
                                RTI                             ;RETURN FROM TRAP
                                .IFTF
```

```
10013                                          ;CLEAR INTERRUPT VECTOR S/R
10014                                          ;
10015                                          ;       JSR     R5,CLRVEC               S/R CALL
10016                                          ;       .WORD   VECT ADR                INT VECT ABS ADR
10017
10018                                          .IFF
10019  040714' 004467  000056          CLRVEC: JSR     R4,SREG                 ;SAVE REGS R0 - R4
10020  040720' 012500                          MOV     (R5)+,R0                ;GET VECTOR ADR
10021  040722' 010001                          MOV     R0,R1
10022  040724' 005721                          TST     (R1)+                   ;SET R1 TO NXT ADR
10023  040726' 010120                          MOV     R1,(R0)+                ;POINT VECTOR AT .+2
10024  040730' 005010                          CLR     (R0)                    ;PUT HALT AT .+2
10025  040732' 000411                          BR      VECEX                   ;GO TO EXIT
10026                                          .IFT
10027                                  CLRVEC: JSR     R4,SREG                 ;SAVE REGISTERS
10028                                          MOV     (R5)+,R0                ;GET VECTOR  ADDRESS
10029                                          MOV     PC,R2                   ;POINT  R2 AT
10030                                          ADD     #VECTBL+6-.,R2          ;INTER PAGE VECTOR TBL
10031                                          MOV     R0,R1
10032                                          ASL     R1                      ;ADJUST R2 TO ROUT ADR ENTRY
10033                                          ADD     R1,R2                   ;FOR THIS VECTOR
10034                                          MOV     R0,R1
10035                                          TST     (R1)+                   ;SET R1 TO NEXT ADR
10036                                          TRAP    6                       ;GO INTO KERN MODE TO CLEAR VECTOR
10037                                                                          ;DOES INST'S AT "KCLRVC" & RET IN-LINE
10038                                          CLR     (R2)                    ;CLEAR INTER-PAGE VECTOR
10039                                          BR      VECEX                   ;GO TO EXIT
10040
10041                                  KCLRVC: MOV     R1,(R0)+                ;POINT VECTOR AT .+2
10042                                          CLR     (R0)                    ;PUT HALT AT .+2
10043                                          RTI                             ;RETURN FROM TRAP
10044                                          .IFTF
10045
10046
10047                                          ;TEST INTERRUPT VECTOR S/R
10048                                          ;
10049                                          ;       MOV     PFWADR(R?),-(SP)        ;PUT PROG'S FLGWD ADR ON STK
10050                                          ;       JSR     R5,TSTVEC               S/R CALL
10051                                          ;       .WORD   VECT ADR                INT VECT ABS ADR
10052                                          ;       .WORD   INTROUT-.               INT ROUT REL ADR
10053                                          ;       BR      LABEL                   EXECUTED IF NOT SAME
10054                                          ;  DESTROYS R0
10055
10056                                          .IFF
10057  040734' 012616                  TSTVEC: MOV     (SP)+,(SP)              ;REMOVE UNUSED FLGWD ADR
10058  040736' 004467  000034                  JSR     R4,SREG                 ;SAVE REGS R0 - R4
10059  040742' 012500                          MOV     (R5)+,R0                ;GET VECTOR ADDR
10060  040744' 010501                          MOV     R5,R1                   ;GET INT ROUT ADR
10061  040746' 062501                          ADD     (R5)+,R1                ;MAKE IT ABSOLUTE
10062  040750' 021001                          CMP     (R0),R1                 ;IS IT SAME AS VECTOR ?
10063  040752' 001001                          BNE     VECEX                   ;NO-RETURN
10064  040754' 005725                          TST     (R5)+                   ;YES-BUMP R5
10065  040756' 004467  000026          VECEX:  JSR     R4,RREG                 ;RESTORE REGS R0 - R4
10066  040762' 000205                          RTS     R5
10067                                          .IFT
10068                                  TSTVEC: MOV     2(SP),R0                ;GET PROG FLGWD ADR OF STK
```

```
10069                                   MOV     (SP)+,(SP)              ;ELIMINATE ITS SPACE ON STK
10070                                   JSR     R4,SREG                 ;SAVE REGISTERS
10071                                   MOV     (R5)+,R1                ;GET VECTOR ADDR
10072                                   MOV     R5,R2                   ;GET INT ROUT ADR
10073                                   ADD     (R5)+,R2                ;MAKE IT REL TO PAGE 0
10074                                   MOV     PC,R3                   ;POINT R3 AT
10075                                   ADD     #VECTBL+6-.,R3          ;INTER-PAGE VECTOR TBL
10076                                   ASL     R1                      ;ADJUST R3 TO ROUT ADR ENTRY
10077                                   ADD     R1,R3                   ;FOR THIS VECTOR
10078                                   BIS     #PSCONS,R2              ;SET PAR5 SELECT BITS IN R2
10079                                   CMP     R2,(R3)                 ;IS VECTOR POINTING AT MY
10080                                   BNE     VECEX                   ;VIRTUAL ADR?  NO-RETURN
10081                                   CMP     R0,-(R3)                ;MY PROG FLGWD MATCH ITS?
10082                                   BNE     VECEX                   ;Y,N-VECEX
10083                                   TST     (R5)+                   ;YES, BUMP R5 & BYPASS BR INST
10084                           VECEX:  JSR     R4,RREG                 ;RESTORE REGS
10085                                   RTS     R5                      ;RETURN
10086
10087
10088                                   ;INTER-PAGE INTERRUPT VECTOR TABLE
10089                                   ;
10090                                   ;
10091
10092                           VECTBL: .REPT   128.
10093                                   JSR     R5,INTCOM               ;TO INTERRUPT  COMMON
10094                                   .WORD   0                       ;SAVED UPARO
10095                                   .WORD   0                       ;INT ROUTINE ADR
10096                                   .ENDR
10097                                   .PAGE
10098                                   ;INTERRUPT INTERCEPT COMMON PROCESSING S/R
10099                                   ;
10100                                   ;
10101
10102                           INTCOM: MOV     @#KPAR5,-(SP)           ;SAVE KERNAL'S PAR 5 &
10103                                   MOV     @#KPAR6,-(SP)           ;PAR 6
10104                                   MOV     SP,PAR6,@#KPAR6         ;POINT PAR 6 AT SHARED CODE PAGE
10105                                   MOV     (R5)+,@#KPAR5           ;SET KERNEL PAR 5 = TO OLD USR PAR 0
10106                                   JSR     PC,@(R5)+               ;GO TO USER'S INTERRUPT ROUT
10107                           INTRET: MOV     (SP)+,@#KPAR6           ;RESTORE KERNEL'S PAR 6 &
10108                                   MOV     (SP)+,@#KPAR5           ;PAR 5
10109                                   MOV     (SP)+,R5                ;RESTORE ORG R5
10110                                   RTI                             ;RETURN FROM INTERRUPT
10111                                   .IFTF
10112
10113
10114                                   ;RETURN FROM INTERRUPT S/R
10115                                   ;
10116                                   ;       JMP     RNTINT                  S/R CALL
10117
10118                                   .IFF
10119  040764' 000002           RTNINT: RTI                             ;RETURN FROM INTERRUPT
10120                                   .IFT
10121                           RTNINT: RTS     PC                      ;RET TO INT COM PROC ROUT AFTER JSR
10122             000                   .ENDC
10123
10124
```

```
10125                                   ;GET DATA BYTE FROM USER'S ABS/VIRT ADR SPACE
10126                                   ;
10127                                   ;       JSR     PC,GETBYT       S/R CALL
10128                                   ;       R0 CONTAINS ABS/VIRT MEMORY ADR
10129                                   ;       RETURNS DATA BYTE IN R1
10130                                   ;       R0 WILL BE INCREMENTED BY 1
10131                                   ;       ALL OTHER REGISTERS ARE UNCHANGED
10132
10133                 001               .IF DF MM
10134                                   .IFF
10135  040766' 112001                   GETBYT: MOVB    (R0)+,R1                ;GET DATA BYTE
10136  040770' 000207                           RTS     PC                      ;EXIT IN-LINE
10137                                   .IFT
10138                                   GETBYT: JSR     R2,GPBCSU               ;DO COMMON SET UP
10139                                           MOVB    (R0),R1                 ;GET DATA BYTE IN R1
10140                                   GPBEX:  MOV     (SP)+,R0                ;RESTORE R0
10141                                           MOV     (SP)+,@#KPDR4           ;RESTORE PDR & PAR REGS
10142                                           MOV     (SP)+,@#KPAR4
10143                                           MOV     (SP)+,R2                ;RESTORE R2
10144                                           INC     R0                      ;ADD 1 TO DATA ADR
10145                                           RTS     PC                      ;EXIT IN-LINE
10146
10147                                   GPBCSU: MOV     @#KPAR4,-(SP)           ;SAVE KERNEL'S PAR 4 &
10148                                           MOV     @#KPDR4,-(SP)           ;PDR 4
10149                                           MOV     R0,-(SP)                ;SAVE ORIG ADR
10150                                           BIC     #17777,R0               ;ISOLATE PAGE SELECT BITS
10151                                           SWAB    R0                      ;COMPUTE DISPL INTO USER'S
10152                                           ASL     R0                      ;PAR/PDR STORAGE
10153                                           ASL     R0
10154                                           ASL     R0
10155                                           ASL     R0
10156                                           SWAB    R0
10157                                           ADD     #P5CONS+PUPARS,R0       ;ADD IN PROG TBL BASE & PAR DISPL
10158                                           MOV     (R0),@#KPAR4            ;LOAD PAR 4 &
10159                                           MOV     16.(R0),@#KPDR4         ;PDR 4
10160                                           MOV     (SP),R0                 ;RESTORE DATA ADR
10161                                           BIC     #160000,R0              ;RESET PAGE SELECT BITS
10162                                           BIS     #P4CONS,R0              ;BASE ADR UPON PAGE 4
10163                                           MOV     R2,PC                   ;EXIT IN-LINE
10164                                   .IFTF
10165
10166
10167                                   ;PUT DATA BYTE IN USER'S ABS/VIRT ADR SPACE
10168                                   ;
10169                                   ;       JSR     PC,PUTBYT       S/R CALL
10170                                   ;       R0 CONTAINS ABS/VIRT MEMORY ADR
10171                                   ;       R1 CONTAINS THE DATA BYTE
10172                                   ;       R0 WILL BE INCREMENTED BY 1
10173                                   ;       ALL OTHER REGISTERS ARE UNCHANGED
10174
10175                                   .IFF
10176  040772' 110120                   PUTBYT: MOVB    R1,(R0)+                ;STORE DATA BYTE
10177  040774' 000207                           RTS     PC                      ;EXIT IN-LINE
10178                                   .IFT
10179                                   PUTBYT: JSR     R2,GPBCSU               ;DO COMMON SET UP
10180                                           MOVB    R1,(R0)                 ;STORE DATA BYTE IN MEMORY
```

```
10181                                   BR      GPBEX                   ;GO TO EXIT
10182                 000               .ENDC
10183
10184
10185                                   ;SAVE REGISTERS R0 THRU R4 S/R
10186                                   ;
10187                                   ;       JSR     R4,SREG         S/R CALL
10188
10189  040776' 010346           SREG:  MOV     R3,-(SP)                ;SAVE REGS R3 - R0
10190  041000' 010246                  MOV     R2,-(SP)                ;R4 SAVED BY JSR TO THIS S/R
10191  041002' 010146                  MOV     R1,-(SP)
10192  041004' 010046                  MOV     R0,-(SP)
10193  041006' 010407                  MOV     R4,PC                   ;EXIT IN-LINE
10194
10195
10196                                   ;RELOAD REGISTERS R0 THRU R4 S/R
10197                                   ;
10198                                   ;       JSR     R4,RREG         S/R CALL
10199
10200  041010' 005726           RREG:  TST     (SP)+                   ;REMOVE THIS JSR'S R4 FROM STK
10201  041012' 012600                  MOV     (SP)+,R0                ;RELOAD R0 - R3
10202  041014' 012601                  MOV     (SP)+,R1
10203  041016' 012602                  MOV     (SP)+,R2
10204  041020' 012603                  MOV     (SP)+,R3
10205  041022' 000204                  RTS     R4                      ;RESTORE ORG R4 & EXIT IN-LINE
```

```
                                .SBTTL  PACKED DECIMAL TO ASCII SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       PACKED DECIMAL TO ASCII SUBROUTINE
                        ;
                        ;       THIS SUBROUTINE CONVERTS FROM 1 TO 6 BCD
                        ;       DIGITS TO EITHER 6 ASCII DIGITS WITH
                        ;       LEADING ZEROES, OR AN EQUIVALENT ASCII NUMBER
                        ;       WITH LEADING BLANKS IGNORED.
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,DECASC
                        ;
                        ;       ENTRY   R0 POINTS TO START OF PACKED CODE
                        ;               R1 POINTS TO START OF RESULT AREA
                        ;
                        ;       EXIT    R0 POINTS ONE BYTE BEYOND PACKED NUMBER
                        ;               R1 POINTS ONE BYTE BEYOND ASCII NUMBER
                        ;
                        ;       DESTROYS R2
                        ;
                        ;*****************************************************************

041024' 010346          DECASC: MOV     R3,-(SP)
041026' 010446                  MOV     R4,-(SP)        ;SAVE HOLY REGISTERS
041030' 012702  000003          MOV     #3,R2           ;SET COUNTER = 3
041034' 112003          LDASLP: MOVB    (R0)+,R3        ;GET TWO PACKED DIGITS
041036' 110304                  MOVB    R3,R4           ;SAVE SECOND DIGIT
041040' 106003                  RORB    R3
041042' 106003                  RORB    R3
041044' 106003                  RORB    R3              ;SHIFT FIRST RIGHT FOUR
041046' 106003                  RORB    R3
041050' 142703  000360          BICB    #360,R3         ;MASK OUT LEFT 4 BITS
041054' 152703  000060          BISB    #60,R3          ;CONVERT TO ASCII
041060' 122703  000077          CMPB    #77,R3          ;CHECK FOR A NULL
041064' 001401                  BEQ     LSKIP1          ;SKIP IF TRUE
041066' 110321                  MOVB    R3,(R1)+        ;STORE ASCII DIGIT
041070' 142704  000360  LSKIP1: BICB    #360,R4         ;MASK LEFT FOUR BITS DIGIT 2
041074' 152704  000060          BISB    #60,R4          ;CONVERT TO ASCII
041100' 122704  000077          CMPB    #77,R4          ;CHECK FOR A NULL
041104' 001401                  BEQ     LSKIP2          ;SKIP IF TRUE
041106' 110421                  MOVB    R4,(R1)+        ;STORE ASCII DIGIT
041110' 005302  LSKIP2: DEC     R2
041112' 001350                  BNE     LDASLP          ;LOOP FOR OTHER DIGITS
041114' 012604                  MOV     (SP)+,R4
041116' 012603                  MOV     (SP)+,R3        ;RESTORE HOLY REGISTERS
041120' 000205                  RTS     R5              ;RETURN TO CALLING PROGRAM
```

```
                                .SBTTL  BINARY TO ASCII SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE CONVERTS ONE OR TWO 16 BIT
                        ;       BINARY NUMBERS INTO 6 OR 8 ASCII DIGITS
                        ;
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,BINASC  (BINAS8)  (BINAS1)
                        ;                               .WORD   RESULT-.
                        ;
                        ;       ENTRY   R0 CONTAINS A 16 BIT UNSIGNED NUMBER
                        ;               R2 CONTAINS UP TO 8 ADDITIONAL BITS OF
                        ;                  HIGHER SIGNIFICANCE THAT WILL BE
                        ;                  CONVERTED IF THE BINAS8 ENTRANCE IS
                        ;                  USED. THESE BITS WILL BE RIGHT JUSTIFIED.
                        ;               RESULT IS THE LABEL OF THE AREA IN WHICH
                        ;               THE ASCII DIGITS ARE TO BE PLACED (6 IF ENTRY
                        ;               AT BINASC, 8 IF AT BINAS8).
                        ;               OUTPUT ADR IS ALREADY IN R1 ON BINAS1 ENTRY.
                        ;       EXIT    R1 CONTAINS ADR OF RIGHTMOST BYTE + 1 IN THE
                        ;               OUTPUT AREA.
                        ;       DESTROYS R0,R2
                        ;
                        ;*****************************************************************

                  001                   .IF DF MM
                                        .IFF
041122'                 BINAS8:
041122' 010501          BINASC: MOV     R5,R1
041124' 062501                  ADD     (R5)+,R1                ;CALCULATE RESULT ADR
041126' 112711  000030  BINAS1: MOVB    #30,(R1)
041132' 000261                  SEC
041134' 006100          LDA1:   ROL     R0                      ;CONVERT TO ASCII
041136' 106121                  ROLB    (R1)+
041140' 111102                  MOVB    (R1),R2
041142' 112711  000206          MOVB    #206,(R1)
041146' 006300          LDA2:   ASL     R0
041150' 001403                  BEQ     LDA3
041152' 106111                  ROLB    (R1)
041154' 103774                  BCS     LDA2
041156' 000766                  BR      LDA1
041160' 110211          LDA3:   MOVB    R2,(R1)
041162' 000205                  RTS     R5                      ;RETURN TO CALLING PGM
                                .IFT
                        BINAS8: MOV     R3,-(SP)                ;SAVE R3
                                MOV     #8.,R3                  ;SET DIGIT CTR TO 8
                                BR      BASCRA                  ;GO CALC RESULT ADDR

                        BINAS1: MOV     R3,-(SP)                ;SAVE R3
                                MOV     #6,R3                   ;SET DIGIT CTR TO 6
                                CLR     R2
                                BR      BASCVT                  ;GO CONVERT

                        BINASC: MOV     R3,-(SP)                ;SAVE R3
```

```
10313                                   MOV     #6,R3           ;SET DIGIT CNTR TO 6
10314                                   CLR     R2
10315                           BASCRA: MOV     R5,R1
10316                                   ADD     (R5)+,R1        ;CALCULATE RESULT ADDRESS
10317                           BASCVT: ADD     R3,R1
10318                                   MOV     R1,-(SP)        ;SAVE RESULT END PTR
10319                                   MOV     R4,-(SP)        ;SAVE R4
10320                           BASLP1: MOV     R0,R4           ;GET BINARY NBR
10321                                   BIC     #177770,R4      ;MASK ALL BUT 3 BITS
10322                                   BISB    #60,R4          ;CONVERT TO ASCII
10323                                   MOVB    R4,-(R1)        ;STORE AS RESULT
10324                                   MOV     #3,R4
10325                           BASLP2: ROR     R2              ;MOVE NEXT 3 BITS
10326                                   ROR     R0              ;INTO POSITION
10327                                   DEC     R4
10328                                   BNE     BASLP2
10329                                   DEC     R3              ;COUNT DIGITS
10330                                   BNE     BASLP1          ;REPEAT FOR ALL
10331                                   MOV     (SP)+,R4        ;RESTORE R4
10332                                   MOV     (SP)+,R1        ;RESTORE RESULT PTR
10333                                   MOV     (SP)+,R3        ;RESTORE R3
10334                                   RTS     R5              ;RETURN
10335           000                     .ENDC
```

```
                                .SBTTL  BINARY TO ASCII DECIMAL SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE CONVERTS ONE 16 BIT BINARY
                        ;       NUMBER INTO ONE TO FIVE ASCII DIGITS
                        ;       REPRESENTING THE DECIMAL VALUE OF THE NUMBER
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,BTASLB       (BTASLZ)
                        ;                               .WORD   RESULT-.
                        ;
                        ;       ENTRY   R0 CONTAINS A 16 BIT UNSIGNED NUMBER
                        ;               RESULT IS THE LABEL OF THE AREA IN
                        ;               WHICH THE ASCII DIGITS ARE TO BE PLACED.
                        ;               THIS AREA MUST BE 6 BYTES & DIGITS ARE
                        ;               RIGHT JUSTIFIED.
                        ;
                        ;               BTASLB IS ENTRY FOR LEADING BLANKS
                        ;               BTASLZ IS ENTRY FOR LEADING ZEROES
                        ;
                        ;       EXIT    R1 POINTS AT LAST BYTE OF RESULT
                        ;
                        ;       DESTROYS R0,R1,R2
                        ;
                        ;*****************************************************************

041164' 012702 000040   BTASLB: MOV     #40,R2          ;STORE SPACE AS FILL
041170' 000402                  BR      LBIASD
041172' 012702 000060   BTASLZ: MOV     #60,R2          ;STORE ZERO AS FILL
041176' 010346          LBIASD: MOV     R3,-(SP)        ;SAVE R3 & R4
041200' 010446                  MOV     R4,-(SP)
041202' 010501                  MOV     R5,R1           ;GET ADR OF RESULT AREA
041204' 062501                  ADD     (R5)+,R1        ;CALCULATE RESULT ADDR.
041206' 005004                  CLR     R4              ;CLEAR ZERO FLAG
041210' 005700                  TST     R0              ;IS NBR ZERO?
041212' 001001                  BNE     LBDSTP          ;Y,N-LBDSTP
041214' 005204                  INC     R4              ;SET THE ZERO FLAG
041216' 010703          LBDSTP: MOV     PC,R3
041220' 062703 000072           ADD     #LBDECTB+2-.,R3 ;SET TABLE POINTER
041224' 110221                  MOVB    R2,(R1)+        ;STORE A FILL CHAR
041226' 112711 000061   LBDLP1: MOVB    #61,(R1)        ;STORE AN ASCII ONE
041232' 161300                  SUB     (R3),R0         ;NUMBER MINUS TBL VALUE
041234' 103414                  BCS     LBDSFL          ;BRANCH IF NEGATIVE
041236' 112702 000060           MOVB    #60,R2          ;SET FILL CHAR = 0
041242' 161300          LBDLP2: SUB     (R3),R0         ;NUMBER MINUS TBL VALUE
041244' 103402                  BCS     LBDDDN          ;DIGIT DONE IF NEG
041246' 105211                  INCB    (R1)            ;INCREMENT RESULT DIGIT
041250' 000774                  BR      LBDLP2          ;AROUND AGAIN
041252' 061300          LBDDDN: ADD     (R3),R0         ;RESTORE POS NBR
041254' 022723 000001           CMP     #1,(R3)+        ;CONVERSION DONE
041260' 001404                  BEQ     LBDTZF          ;YES-GO CHECK ZERO FLAG
041262' 005201                  INC     R1              ;BUMP OUTPUT POINTER
041264' 000760                  BR      LBDLP1          ;GO WORK ON NEXT DIGIT
041266' 110211          LBDSFL: MOVB    R2,(R1)         ;STORE THE FILL CHAR
041270' 000770                  BR      LBDDDN          ;CHECK IF FINISHED
```

```
10393  041272' 005704                 LBDTZF: TST     R4                      ;TEST THE ZERO FLAG
10394  041274' 001402                         BEQ     LBDEXT                  ;IF NOT SET, EXIT
10395  041276' 112711  000060                 MOVB    #60,(R1)                ;ELSE INSERT A ZERO
10396  041302' 012604                 LBDEXT: MOV     (SP)+,R4                ;RESTORE R3 & R4
10397  041304' 012603                         MOV     (SP)+,R3
10398  041306' 000205                         RTS     R5                      ;RETURN
10399
10400  041310' 177777                 LDECTB: .WORD   65535.                  ;DECIMAL TO BINARY TABLE
10401  041312' 023420                         .WORD   10000.
10402  041314' 001750                         .WORD   1000.
10403  041316' 000144                         .WORD   100.
10404  041320' 000012                         .WORD   10.
10405  041322' 000001                         .WORD   1
```

```
10407                                           .SBTTL  RANDOM NUMBER GENERATOR ROUTINE
10408
10409                                   ;*****************************************************************
10410                                   ;
10411                                   ;*      JSR     R5,LRANDM               ;CALL THE ROUTINE
10412                                   ;*      RETURN                          ;RETURN HERE THE RANDOM
10413                                   ;*                                      ;NUMBER WILL BE IN
10414                                   ;*                                      ;LHINUM,LLONUM
10415                                   ;
10416                                   ;*****************************************************************
10417
10418
10419  041324' 010346                   LRANDM: MOV     R3,-(SP)                ;SAVE R3
10420  041326' 016700  000104                   MOV     LLONUM,R0               ;SET R0 WITH LOW
10421  041332' 016701  000076                   MOV     LHINUM,R1               ;SET R1 WITH HIGH
10422  041336' 012703  177771                   MOV     #-7,R3                  ;SET SHIFT COUNT
10423  041342' 005002                           CLR     R2                      ;ZERO R2
10424  041344' 006300                   1$:     ASL     R0                      ;SHIFT R0 $EFT AND
10425  041346' 006101                           ROL     R1                      ;ROTATE CARRY INTO R1 AND
10426  041350' 006102                           ROL     R2                      ;ROTATE CARRY INTO R2
10427  041352' 005203                           INC     R3                      ;CHECK FOR DONE
10428  041354' 001373                           BNE     1$                      ;CONTINUE SHIFT $OOP
10429  041356' 066700  000054                   ADD     LLONUM,R0               ;ADD NUMBER TO MAKE X 129
10430  041362' 005501                           ADC     R1                      ;PROPOGATE CARRY
10431  041364' 066701  000044                   ADD     LHINUM,R1               ;ADD NUMBER TO MAKE X 129
10432  041370' 005502                           ADC     R2                      ;PROPOGATE CARRY
10433  041372' 062700  001057                   ADD     #1057,R0                ;ADD LOW CONSTANT
10434  041376' 005501                           ADC     R1                      ;PROPOGATE CARRY
10435  041400' 005502                           ADC     R2                      ;PROPOGATE CARRY
10436  041402' 062701  047401                   ADD     #47401,R1               ;ADD HIGH CONSTANT
10437  041406' 005502                           ADC     R2                      ;PROPOGATE CARRY
10438  041410' 062702  000006                   ADD     #6,R2                   ;ADD HIGHEST CONSTART
10439  041414' 060200                           ADD     R2,R0                   ;REPRIME R0 WITH HIGHEST DIGIT
10440  041416' 005501                           ADC     R1                      ;PROPOGATE CARRY
10441  041420' 010067  000012                   MOV     R0,LLONUM               ;SAVE R0
10442  041424' 010167  000004                   MOV     R1,LHINUM               ;SAVE R1
10443  041430' 012603                           MOV     (SP)+,R3                ;RESTORE R3
10444  041432' 000205                           RTS     R5                      ;RETURN
10445  041434' 176543                   LHINUM: .WORD   176543
10446  041436' 123456                   LLONUM: .WORD   123456
```

```
10448                                           .SBTTL  DATA BUFFERS, COMMON WORDS, & PATTERN ADDRESES TABLE
10449
10450                                           .EVEN
10451  041440' 000000  000000  000000  CBUF00: .WORD   0,0,0,0,0,0,0,0        ;COMMON DATA BUFFERS
       041446' 000000  000000  000000
       041454' 000000  000000
10452  041460' 000000  000000  000000  CBUF01: .WORD   0,0,0,0,0,0,0,0
       041466' 000000  000000  000000
       041474' 000000  000000
10453  041500' 000000  000000  000000  CBUF02: .WORD   0,0,0,0,0,0,0,0
       041506' 000000  000000  000000
       041514' 000000  000000
10454  041520' 000000  000000  000000  CBUF03: .WORD   0,0,0,0,0,0,0,0
       041526' 000000  000000  000000
       041534' 000000  000000
10455  041540' 000000  000000  000000  CBUF04: .WORD   0,0,0,0,0,0,0,0
       041546' 000000  000000  000000
       041554' 000000  000000
10456  041560' 000000  000000  000000  CBUF05: .WORD   0,0,0,0,0,0,0,0
       041566' 000000  000000  000000
       041574' 000000  000000
10457  041600' 000000  000000  000000  CBUF06: .WORD   0,0,0,0,0,0,0,0
       041606' 000000  000000  000000
       041614' 000000  000000
10458  041620' 000000  000000  000000  CBUF07: .WORD   0,0,0,0,0,0,0,0
       041626' 000000  000000  000000
       041634' 000000  000000
10459  041640' 000000  000000  000000  CBUF08: .WORD   0,0,0,0,0,0,0,0
       041646' 000000  000000  000000
       041654' 000000  000000
10460  041660' 000000  000000  000000  CBUF09: .WORD   0,0,0,0,0,0,0,0
       041666' 000000  000000  000000
       041674' 000000  000000
10461  041700' 000000  000000  000000  CBUF10: .WORD   0,0,0,0,0,0,0,0
       041706' 000000  000000  000000
       041714' 000000  000000
10462  041720' 000000  000000  000000  CBUF11: .WORD   0,0,0,0,0,0,0,0
       041726' 000000  000000  000000
       041734' 000000  000000
10463  041740' 000000  000000  000000  CBUF12: .WORD   0,0,0,0,0,0,0,0
       041746' 000000  000000  000000
       041754' 000000  000000
10464  041760' 000000  000000  000000  CBUF13: .WORD   0,0,0,0,0,0,0,0
       041766' 000000  000000  000000
       041774' 000000  000000
10465  042000' 000000  000000  000000  CBUF14: .WORD   0,0,0,0,0,0,0,0
       042006' 000000  000000  000000
       042014' 000000  000000
10466  042020' 000000  000000  000000  CBUF15: .WORD   0,0,0,0,0,0,0,0
       042026' 000000  000000  000000
       042034' 000000  000000
```

```
10468  042040' 000000          LCOM0:  .WORD   0                       ;COMMON DATA WORD STORAGE
10469  042042' 000000          LCOM1:  .WORD   0
10470  042044' 000000          LCOM2:  .WORD   0
10471  042046' 000000          LCOM3:  .WORD   0
10472  042050' 000000          LCOM4:  .WORD   0
10473  042052' 000000          LCOM5:  .WORD   0
10474  042054' 000000          LCOM6:  .WORD   0
10475  042056' 000000          LCOM7:  .WORD   0
10476  042060' 000000          LCOM8:  .WORD   0
10477  042062' 000000          LCOM9:  .WORD   0
10478
10479
10480  042064' 000004          LPAT0P: .WORD   LPAT0-LPBASE             ;PATTERN ADDRESS TABLE
10481  042066' 000036          LPAT1P: .WORD   LPAT1-LPBASE
10482  042070' 000070          LPAT2P: .WORD   LPAT2-LPBASE
10483  042072' 000120          LPAT3P: .WORD   LPAT3-LPBASE
10484  042074' 000172          LPAT4P: .WORD   LPAT4-LPBASE
10485  042076' 000204          LPAT5P: .WORD   LPAT5-LPBASE
10486  042100' 000216          LPAT6P: .WORD   LPAT6-LPBASE
10487  042102' 000230          LPAT7P: .WORD   LPAT7-LPBASE
10488  042104' 000242          LPAT8P: .WORD   LPAT8-LPBASE
10489  042106' 000254          LPAT9P: .WORD   LPAT9-LPBASE
10490  042110' 000266          LPATAP: .WORD   LPATA-LPBASE
10491  042112' 000314          LPATBP: .WORD   LPATB-LPBASE
10492          042114'          LPATEN= .
10493
10494  042114' 005015          LCRETN: .WORD   5015
10495
10496
10497          042116'          PG6END= .               ;CODE BEYOND THIS POINT IS NOT REQUIRED TO BE IN
10498                                                   ;THE SHARED PAGE AREA.
10499
10500
10501              001                 .IF NDF MIMIC
10502              002                 .IF DF MM
10503                                  .IFF
10504                                  .TITLE  MAINDEC-11-DTMGA-C   MPG CONTROL ROUTINE
10505                                  .IFT
10506                                  .TITLE  MAINDEC-11-DTMMA-B   MPG CONTROL ROUTINE
10507              001                 .ENDC
10508              000                 .ENDC
10509              001                 .IF DF MIMIC
10510              002                 .IF DF MM
10511                                  .IFF
10512                                  .TITLE  MAINXEC-11-DTMGA-C   MPG CONTROL ROUTINE
10513                                  .IFT
10514                                  .TITLE  MAINXEC-11-DTMMA-B   MPG CONTROL ROUTINE
10515              001                 .ENDC
10516              000                 .ENDC
```

```
                                .SBTTL  DECIMAL OR OCTAL TO BINARY SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE CONVERTS FROM 1 TO 6 PACKED BCD DIGITS (3
                        ;       BYTES), THAT REPRESENT OCTAL OR DECIMAL NUMBERS, INTO ONE
                        ;       16 BIT BINARY NUMBER.  ANY NUMBER LARGER THAN
                        ;       65,535 WILL CAUSE AN ERROR EXIT.
                        ;
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,DECBIN       (OCTBIN)
                        ;                               .WORD   ERRTN-.
                        ;
                        ;       ENTRY   R0 CONTAINS ADDRESS OF DECIMAL NBR (FIRST BTYE)
                        ;               ERRTN IS THE ERROR RETURN ADDRESS TAG
                        ;
                        ;       EXIT    R0  CONTAINS RESULT
                        ;
                        ;       DESTROYS R2
                        ;
                        ;*****************************************************************

042116' 010702          DECBIN: MOV     PC,R2                   ;SET POINTER R2
042120' 062702  177170          ADD     #LDECTB-.,R2            ;TO DECIMAL TABLE
042124' 010146                  MOV     R1,-(SP)                ;SAVE R1
042126' 005001                  CLR     R1                      ;CLEAR OCTAL FLAG
042130' 000406                  BR      LTBSAV                  ;GO SAVE REGS
042132' 010702          OCTBIN: MOV     PC,R2                   ;SET POINTER R2
042134' 062702  000172          ADD     #LOCTTB-.,R2            ;TO OCTAL TABLE
042140' 010146                  MOV     R1,-(SP)                ;SAVE R1
042142' 012701  000001          MOV     #1,R1                   ;SET OCTAL FLAG
042146' 010346          LTBSAV: MOV     R3,-(SP)
042150' 010446                  MOV     R4,-(SP)                ;SAVE HOLY REGISTERS
042152' 010546                  MOV     R5,-(SP)
042154' 005003                  CLR     R3                      ;CLEAR RESULT WORD
042156' 012704  000006          MOV     #6,R4                   ;SET UP DIGIT COUNT
042162' 111005          LTBIN0: MOVB    (R0),R5                 ;ALIGN BYTE TO GET
042164' 106005                  RORB    R5                      ;FIRST DIGIT IN BYTE
042166' 106005                  RORB    R5
042170' 106005                  RORB    R5
042172' 106005                  RORB    R5
042174' 000401                  BR      LTBIN2                  ;GO PROCESS DIGIT
042176' 112005          LTBIN1: MOVB    (R0)+,R5                ;GET INPUT BYTE AGAIN
042200' 042705  177760  LTBIN2: BIC     #177760,R5              ;RESET ANY OTHER BITS
042204' 005701                  TST     R1                      ;BRANCH IF OCTAL FLAG
042206' 001406                  BEQ     LTBFD                   ;NOT SET
042210' 120527  000010          CMPB    R5,#8.                  ;CHECK IF DECIMAL NBR
042214' 001432                  BEQ     LTBERR                  ;ERROR IF TRUE
042216' 120527  000011          CMPB    R5,#9.                  ;CHECK IF DECIMAL NBR
042222' 001427                  BEQ     LTBERR                  ;ERROR IF TRUE
042224' 122705  000017  LTBFD:  CMPB    #17,R5                  ;ALL ONES ?
042230' 001406                  BEQ     LTBIN4                  ;TRY NEXT DIGIT
042232' 105705                  TSTB    R5                      ;ALL ZEROES ?
042234' 001404                  BEQ     LTBIN4                  ;TRY NEXT DIGIT
042236' 061203          LTBIN3: ADD     (R2),R3                 ;ADD TABLE VALUE TO RESULT
```

```
10574  042240' 103420                  BCS     LTBERR                  ;ERROR IF CARRY SET
10575  042242' 005305                  DEC     R5
10576  042244' 001374                  BNE     LTBIN3                  ;REPEAT UNTIL DIGIT = 0
10577  042246' 005722          LTBIN4: TST     (R2)+                   ;BUMP TABLE POINTER
10578  042250' 005304                  DEC     R4                      ;DECR DIGIT CNT
10579  042252' 001404                  BEQ     LTBIN5                  ;DONE 6 DIGITS? (N,Y-LTBIN5)
10580  042254' 032704 000001           BIT     #1,R4                   ;2ND DIGIT OF BYTE NEXT?
10581  042260' 001346                  BNE     LTBIN1                  ;N,Y-LTBIN1
10582  042262' 000737                  BR      LTBIN0                  ;DO 1ST DIGIT OF NXT BYTE
10583  042264' 114004          LTBIN5: MOVB    -(R0),R4                ;GET BYTE WITH LAST DIGIT
10584  042266' 042704 177760           BIC     #177760,R4              ;ISOLATE LAST DIGIT
10585  042272' 001006                  BNE     LTBOUT                  ;WAS IT ZERO? (Y,N-LTBOUT)
10586  042274' 010304                  MOV     R3,R4                   ;MOVE RESULT TO  A WORK REG
10587  042276' 005204                  INC     R4                      ;INCREMENT RESULT
10588  042300' 103003                  BCC     LTBOUT                  ;ERROR IF CARRY GETS SET
10589  042302' 012605          LTBERR: MOV     (SP)+,R5                ;RESTORE R5
10590  042304' 061505                  ADD     (R5),R5                 ;CALCULATE ERR RET ADR
10591  042306' 000403                  BR      LBTEX                   ;GO TO S/R EXIT
10592  042310' 010300          LTBOUT: MOV     R3,R0                   ;MOVE RESULT TO RETURN REG
10593  042312' 012605                  MOV     (SP)+,R5                ;RESTORE R5
10594  042314' 005725                  TST     (R5)+                   ;BYPASS ERR RET ADR
10595  042316' 012604          LBTEX:  MOV     (SP)+,R4                ;RESTORE HOLY REGISTERS
10596  042320' 012603                  MOV     (SP)+,R3
10597  042322' 012601                  MOV     (SP)+,R1
10598  042324' 000205                  RTS     R5                      ;RETURN TO CALLING PGM
10599
10600  042326' 100000          LOCTTB: .WORD   100000                  ;OCTAL TO BINARY TABLE
10601  042330' 010000                  .WORD   10000
10602  042332' 001000                  .WORD   1000
10603  042334' 000100                  .WORD   100
10604  042336' 000010                  .WORD   10
10605  042340' 000001                  .WORD   1
10606                   001             .IF DF MM
10607                                   .PAGE
10608                                   .SBTTL  ASCII OCTAL TO BINARY SUBROUTINE
10609
10610                           ;*****************************************************************
10611                           ;
10612                           ;       THIS SUBROUTINE CONVERTS FROM 1 TO 8 ASCII
10613                           ;       CHARACTERS THAT REPRESENT THE NUMBERS FROM
10614                           ;       0 THRU 7, TO THE EQUIVALENT BINARY NUMBER.
10615                           ;       NUMBERS  GREATER THAN 8 DIGITS OR NON-NUMERIC
10616                           ;       CHARACTERS WILL CAUSE AN ERROR EXIT.
10617                           ;
10618                           ;       CALLING SEQUENCE        JSR     R5,ASOCBN
10619                           ;                               .WORD   ERRTN-.
10620                           ;
10621                           ;       ENTRY   R0 CONTAINS ADDRESS OF  LAST DIGIT
10622                           ;               R1 CONTAINS A  COUNT OF THE NUMBER OF DIGITS
10623                           ;               ERRTN IS THE ERROR RETURN ADDRESS TAG
10624                           ;
10625                           ;       EXIT    R0 CONTAINS LSB OF RESULT
10626                           ;               R2 CONTAINS MSB OF RESULT
10627                           ;
10628                           ;*****************************************************************
10629
```

```
10630
10631                           ASOCBN: MOV     R3,-(SP)        ;SAVE R3 AND R4
10632                                   MOV     R4,-(SP)
10633                                   CMP     R1,#8.          ;HOW MANY DIGITS?
10634                                   BLE     LASOB2          ;BR IF 8 OR LESS DIGITS
10635                                   MOV     R0,R3           ;CK FOR & BYPASS LEADING 0'S
10636                                   SUB     R1,R3           ;SET POINTER R3
10637                                   INC     R3              ;TO FIRST DIGIT
10638                           LASOB1: CMPB    (R3)+,#60       ;IS IT A ZERO?
10639                                   BNE     LASOBE          ;NO-ERROR
10640                                   DEC     R1              ;IS NEXT DIGIT
10641                                   CMP     R1,#8.          ;THE FIRST OF 8?
10642                                   BGT     LASOB1          ;NO-CONTINUE CHECK
10643                           LASOB2: MOV     #9.,R4          ;COMPUTE # OF FILL DIGITS + 1
10644                                   SUB     R1,R4
10645                                   MOV     R0,R3           ;PUT DATA ADR IN WORK REG
10646                                   CLR     R2              ;CLEAR HIGH ORDER BIT REG
10647                           LASOB3: CMPB    (R3),#60        ;IF LESS THAN ASCII 0
10648                                   BLO     LASOBE          ;TAKE ERROR EXIT
10649                                   CMPB    (R3),#67        ;IF MORE THAN ASCII 7
10650                                   BHI     LASOBE          ;TAKE ERROR EXIT
10651                                   MOVB    (R3),-(SP)      ;ABSORB THE DIGIT
10652                                   BICB    #370,(SP)       ;EXTRACT OCTAL DIGIT BITS
10653                                   JSR     PC,LASHFT       ;SHIFT THEM INTO THE RESULT
10654                                   DEC     R3              ;POINT TO NEXT CHAR
10655                                   DEC     R1              ;IF MORE DIGITS TO GO
10656                                   BNE     LASOB3          ;GO GET THEM
10657                           LASOB4: DEC     R4              ;DECR FILL DIGIT CNT
10658                                   BEQ     LASOB5          ;ANY MORE TO DO? BR IF NO
10659                                   CLR     -(SP)           ;SET UP FILL DIGIT BITS
10660                                   JSR     PC,LASHFT       ;SHIFT IN FILL DIGIT
10661                                   BR      LASOB4          ;GO CK FOR MORE
10662                           LASOB5: TST     (R5)+           ;BYPASS ERR RET ADR
10663                           LASOB6: MOV     (SP)+,R4        ;RESTORE R4 AND R3
10664                                   MOV     (SP)+,R3
10665                                   RTS     R5              ;EXIT
10666
10667                           LASOBE: ADD     (R5),R5         ;CALCULATE ERROR RETN
10668                                   BR      LASOB6          ;GO TO SUBR EXIT
10669
10670
10671                           LASHFT: MOV     #3,-(SP)        ;SET UP LOOP CNT
10672                           LASHF1: ROR     4(SP)           ;SHIFT BIT OUT OF OCTAL DIGIT
10673                                   RORB    R2              ;SHIFT IT INTO RESULT REGS
10674                                   ROR     R0
10675                                   DEC     (SP)            ;DONE 3 BITS?
10676                                   BNE     LASHF1          ;Y,N-LASHF1
10677                                   TST     (SP)+           ;REMOVE LOOP CNT FROM STK
10678                                   MOV     (SP)+,(SP)      ;REMOVE DIGIT FROM STK
10679                                   RTS     PC              ;EXIT IN-LINE
10680                   000             .ENDC
```

```
                                .SBTTL  ASCII TO PACKED DECIMAL SUBROUTINE

                        ;*****************************************************************
                        ;
                        ;       THIS SUBROUTINE CONVERTS FROM 1 TO 6 ASCII
                        ;       CHARACTERS THAT REPRESENT THE NUMBERS FROM
                        ;       0 THRU 9, TO THE EQUIVALENT PACKED DECIMAL
                        ;       FORMAT, AND STORES THEM IN AN INTERNAL BUFFER
                        ;       CALLED DECBUF.  NUMBERS GREATER THAN 6 DIGITS
                        ;       OR NON-NUMERIC CHARACTERS WILL CAUSE AN ERROR EXIT.
                        ;
                        ;       CALLING SEQUENCE        JSR     R5,ASDECO   (ASDEC1)
                        ;                               .WORD   ERRTN-.
                        ;
                        ;       ENTRY   R0 CONTAINS ADDRESS OF LAST DIGIT
                        ;               R1 CONTAINS A COUNT OF THE NUMBER OF DIGITS.
                        ;               ERRTN IS THE ERROR RETURN ADDRESS TAG
                        ;               ASDECO FILLS LEADING DIGITS WITH ZEROES
                        ;               ASDEC1 FILLS LEADING DIGITS WITH ONES
                        ;
                        ;       EXIT    DECBUF CONTAINS PACKED NUMBER (3 BYTES)
                        ;               R0 POINTS TO DECBUF
                        ;
                        ;       DESTROYS R1,R2
                        ;
                        ;*****************************************************************

042342' 012702 177777   ASDEC1: MOV     #177777,R2      ;SET UP FILL CHAR OF ALL ONES
042346' 000401                   BR      LASDEC
042350' 005002          ASDECO: CLR     R2              ;SET UP FILL CHAR OF 0'S
042352' 010346          LASDEC: MOV     R3,-(SP)
042354' 010446                  MOV     R4,-(SP)        ;SAVE HOLY REGISTERS
042356' 020127 000006           CMP     R1,#6
042362' 003412                  BLE     LASCNT          ;CONTINUE IF 6 OR LESS DIGITS
042364' 010003                  MOV     R0,R3           ;CK FOR & BYPASS LEADING 0'S
042366' 160103                  SUB     R1,R3           ;SET POINTER R3
042370' 005203                  INC     R3              ;TO FIRST DIGIT
042372' 122327 000060   LASLP0: CMPB    (R3)+,#60       ;IS IT A ZERO ?
042376' 001052                  BNE     LASER1          ;NO-ERROR
042400' 005301                  DEC     R1              ;IS NEXT DIGIT
042402' 020127 000006           CMP     R1,#6           ;THE FIRST OF 6 ?
042406' 003371                  BGT     LASLP0          ;NO - CONTINUE CHECK
042410' 012704 000006   LASCNT: MOV     #6,R4           ;COMPUTE # OF FILL CHAR'S
042414' 160104                  SUB     R1,R4
042416' 010603                  MOV     SP,R3           ;SAVE STK PNTR IF AN ERR
042420' 121027 000060   LASLP1: CMPB    (R0),#60        ;IF LESS THAN ASCII 0
042424' 103436                  BLO     LASERR          ;TAKE ERROR EXIT
042426' 121027 000071           CMPB    (R0),#71        ;IF MORE THAN ASCII 9
042432' 101033                  BHI     LASERR          ;TAKE ERROR EXIT
042434' 111046                  MOVB    (R0),-(SP)      ;ABSORB THE DIGIT
042436' 005300                  DEC     R0              ;POINT TO NXT CHAR
042440' 005301                  DEC     R1              ;IF MORE DIGITS TO GO
042442' 001366                  BNE     LASLP1          ;GO GET THEM
```

```
10738  042444' 005704                      TST     R4              ;NEED FILL CHAR'S?
10739  042446' 001403                      BEQ     LASLP3          ;Y,N-LASLP3
10740  042450' 010246              LASLP2: MOV     R2,-(SP)        ;STORE A FILL CHAR
10741  042452' 005304                      DEC     R4              ;DECR FILL CHAR CNT
10742  042454' 001375                      BNE     LASLP2          ;DONE ALL? (Y,N-LASLP2)
10743  042456' 004767  000046      LASLP3: JSR     PC,LASPAK       ;PACK 1ST TWO DIGITS
10744  042462' 010401                      MOV     R4,R1           ;SAVE THE BYTE
10745  042464' 004767  000040              JSR     PC,LASPAK       ;PACK NXT 2 DIGITS
10746  042470' 010402                      MOV     R4,R2           ;SAVE THEM
10747  042472' 004767  000032              JSR     PC,LASPAK       ;PACK LAST 2 DIGITS
10748  042476' 010700                      MOV     PC,R0           ;SET UP ADR OF DECBUF
10749  042500' 062700  000065              ADD     #DECBUF+2-.,R0
10750  042504' 110410                      MOVB    R4,(R0)         ;STORE 3 BYTES IN
10751  042506' 110240                      MOVB    R2,-(R0)        ;DECBUF
10752  042510' 110140                      MOVB    R1,-(R0)
10753  042512' 005725                      TST     (R5)+           ;BYPASS ERR RET ADR
10754  042514' 012604              LASEX:  MOV     (SP)+,R4        ;RESTORE HOLY REGISTERS
10755  042516' 012603                      MOV     (SP)+,R3
10756  042520' 000205                      RTS     R5              ;RETURN TO CALLING PGM
10757  042522' 010306              LASERR: MOV     R3,SP           ;RESTORE STACK PNTR
10758  042524' 061505              LASER1: ADD     (R5),R5         ;CALCULATE ERROR RETURN
10759  042526' 000772                      BR      LASEX           ;GO TO S/R EXIT
10760
10761
10762  042530' 012600              LASPAK: MOV     (SP)+,R0        ;SAVE RET ADR
10763  042532' 112603                      MOVB    (SP)+,R3        ;GET BYTE WITH LEFT FOUR BITS
10764  042534' 042703  177760              BIC     #177760,R3      ;CLEAR ANY OTHER BITS
10765  042540' 106103                      ROLB    R3              ;SHIFT THEM LEFT FOUR BITS
10766  042542' 106103                      ROLB    R3
10767  042544' 106103                      ROLB    R3
10768  042546' 106103                      ROLB    R3
10769  042550' 012604                      MOV     (SP)+,R4        ;GET BYTE WITH RIGHT FOUR BITS
10770  042552' 042704  177760              BIC     #177760,R4      ;CLEAR ANY OTHER BITS
10771  042556' 050304                      BIS     R3,R4           ;MERGE 2 DIGITS INTO 1 BYTE
10772  042560' 010007                      MOV     R0,PC           ;EXIT IN-LINE
10773
10774
10775  042562' 000001                      .BLKB   1
10776  042563' 000003              DECBUF: .BLKB   3
```

```
                                .SBTTL  VALID DEVICES TABLE


                                ;THE FOLLOWING TABLE WILL BE READ INTO MEMORY BY
                                ;THE ONE TIME HOUSEKEEPING ROUTINE. ITS FILENAME
                                ;ON THE MPG MEDIUM IS "TVDA??.MPG".


                                VALID DEVICE MODEL NAME TABLE

                                    6 BYTE ENTRY FORMAT:

                                        BYTES 0 THRU 3 = 4 CHAR ASCII DEV NAME

                                        BYTE 4 = FLAG BYTE
                                          BIT 0 = 1 = VALID LIST DEV
                                          BIT 1 = 1 = VALID SAVE DEV
                                          BIT 2 = 1 = VALID FETCH DEV
                                          BIT 3 = 1 = PERM SAV/FTH DEV
                                          BIT 4 = DEVICE I.D. BITS: USED TO
                                            & 5 = DISTINGUISH BETWEEN DIFFERENT
                                            & 6 = MODEL NAMES AND ALSO BETWEEN
                                            & 7 = DEVICES USING SAME DEV ROUT.

                                        BYTE 5 = POINTER INTO THE DEVICE
                                                 ROUTINE I.D. CODE TBL FOR XX
                                                 (TXXANN.MPG) OR, IF ODD, THE
                                                 I.D. CODE FOR 'NONE', 'KYBD',
                                                 'KBDJ', OR 'LOAD'


                                                 NOTE: BITS 1 & 2 OF BYTE 4 WILL
                                                       BE RESET BY MPG'S 1 TIME HSKP
                                                       FOR ALL DEVICES EXCEPT THE
                                                       LOAD DEV & PAPER TAPE.

                    001                         .IF DF MIMIC
                                                .IFF
042566' 000000      VDFLSZ: .WORD   0                       ;DUMMY ALLOCATIONS USED TO
042570' 000000      DIDTAD: .WORD   0                       ;PROPERLY ALIGN LABELS IN
042572'             CMDLNM:                                 ;THIS AREA.
042572' 000003      CNONID: .BLKW   3
042600' 000003      CKBDID: .BLKW   3
042606' 000003              .BLKW   3
042614' 000003      CLDID:  .BLKW   3

                    043566'             .=      VDFLSZ+512.         ;ALLOW FOR UP TO 512 BYTE TABLE

                                        ;MPG WILL SET THE USER PROGRAM AREA ADDRESS
                                        ;TO THE END OF THE TABLE THAT WAS READ IN.
                                                .IFT
                    VDFLSZ: .WORD   CTBEND-.                ;NUMBER OF BYTES IN THIS FILE
                    DIDTAD: .WORD   DRIDCD-.                 ;RELATIVE ADR OF I.D. TABLE


                    CMDLNM:
                    CNONID: .ASCII  /NONE/<377><001.>
```

```
10834                                   CKBDID: .ASCII  /KYBD/<377><003.>
10835                                           .ASCII  /KBDD/<377><003.>
10836                                   CLDID:  .ASCII  /LOAD/<377><005.>
10837                                           .ASCII  /TC11/<006><000.>
10838                                           .ASCII  /TU56/<026><000.>
10839                                           .ASCII  /RK11/<006><002.>
10840                                           .ASCII  /RK05/<026><002.>
10841                                           .ASCII  /LP11/<001><004.>
10842                                           .ASCII  /LS11/<021><004.>
10843                                           .ASCII  /LV11/<041><004.>
10844                                           .ASCII  /TM11/<006><006.>
10845                                           .ASCII  /TU10/<026><006.>
10846                                           .ASCII  /DJ11/<000><008.>
10847                                           .ASCII  /DL11/<000><010.>
10848                                           .ASCII  /DQ11/<000><012.>
10849                                           .ASCII  /PC11/<016><014.>
10850                                           .ASCII  /PR11/<034><014.>
10851                                           .ASCII  /RP04/<006><016.>
10852                                           .ASCII  /RP05/<026><016.>
10853                                           .ASCII  /RP06/<046><016.>
10854                                           .ASCII  /DH11/<000><018.>
10855                                           .ASCII  /TA11/<000><020.>
10856                                           .ASCII  /RF11/<000><022.>
10857                                           .ASCII  /RC11/<000><024.>
10858                                           .ASCII  /RP02/<000><026.>
10859                                           .ASCII  /RP03/<020><026.>
10860                                           .ASCII  /DN11/<000><028.>
10861                                           .ASCII  /DC11/<000><030.>
10862                                           .ASCII  /CR11/<000><032.>
10863                                           .ASCII  /CM11/<020><032.>
10864                                           .ASCII  /CD11/<000><034.>
10865                                           .ASCII  /KW11/<000><036.>
10866                                           .ASCII  /TM02/<006><038.>
10867                                           .ASCII  /TU16/<026><038.>
10868                                           .ASCII  /RX11/<006><040.>
10869                                           .ASCII  /RX01/<026><040.>
10870                                           .ASCII  /DU11/<000><042.>
10871                                           .ASCII  /RK06/<000><044.>
10872                                           .ASCII  /RS03/<000><046.>
10873                                           .ASCII  /RS04/<020><046.>
10874                                           .ASCII  /RL01/<000><048.>
10875                                           .ASCII  /RL02/<020><048.>
10876                                           .ASCII  /????/<377><377>        ;TABLE TERMINATOR
10877
10878
10879                                           ;VALID DEV ROUTINE I.D. TABLE
10880
10881                                   DRIDCD: .ASCII  /TC/                    ;TC11/TU56
10882                                           .ASCII  /RK/                    ;RK11/RK05
10883                                           .ASCII  /LP/                    ;LP11/LS11/LV11
10884                                           .ASCII  /TM/                    ;TM11/TU10
10885                                           .ASCII  /DJ/                    ;DJ11
10886                                           .ASCII  /DL/                    ;DL11
10887                                           .ASCII  /DQ/                    ;DQ11
10888                                           .ASCII  /PC/                    ;PC11/PR11
10889                                           .ASCII  /RP/                    ;RP04/RP05/RP06
```

```
10890                                   .ASCII  /DH/                   ;DH11
10891                                   .ASCII  /MS/                   ;TA11
10892                                   .ASCII  /MS/                   ;RF11
10893                                   .ASCII  /MS/                   ;RC11
10894                                   .ASCII  /R3/                   ;RP02/RP03
10895                                   .ASCII  /MS/                   ;DN11
10896                                   .ASCII  /MS/                   ;DC11
10897                                   .ASCII  /MS/                   ;CR11/CM11
10898                                   .ASCII  /MS/                   ;CD11
10899                                   .ASCII  /MS/                   ;KW11
10900                                   .ASCII  /MS/                   ;TM02/TU16
10901                                   .ASCII  /MS/                   ;RX11/RX01
10902                                   .ASCII  /DU/                   ;DU11
10903                                   .ASCII  /R6/                   ;RK06
10904                                   .ASCII  /RS/                   ;RS03/RS04
10905                                   .ASCII  /RL/                   ;RL01/RL02
10906                           CTBEND  =.
10907           000                     .ENDC
```

```
10909                                   .SBTTL  MPG CONTROL ROUTINE ONE TIME HOUSEKEEPING
10910
10911                                   ;*************************************************************************
10912                                   ;
10913                                   ;       THIS CODING WILL BE EXECUTED ONCE WHEN
10914                                   ;       MPG IS FIRST LOADED. IT WILL BE DESTROYED
10915                                   ;       WHEN THE FIRST USER PROGRAM IS ENTERED OR
10916                                   ;       FETCHED.
10917                                   ;
10918                                   ;*************************************************************************
10919
10920
10921                                           ;R0 CONTAINS THE ADDRESS OF THE END OF MEMORY AS
10922                                           ;COMPUTED BY THE EXEC. (ALWAYS < 32K WORDS)
10923                                           ;R2 CONTAINS THE POSITION ADJUST FACTOR FOR THE
10924                                           ;EXEC'S GLOBAL TABLE. IT WILL BE 0'S ON THE
10925                                           ;NON-MEM MGMNT VERSION.
10926
10927
10928                                           ;STORE MEM END ADR & HSKP INTERRUPT VECTORS
10929
10930  043566' 010067  135660   C1THKP: MOV     R0,CRMEND               ;STORE END OF REAL MEMORY ADR
10931  043572' 012703  000300           MOV     #300,R3                 ;GET VECTOR'S START ADR
10932  043576' 010304                   MOV     R3,R4                   ;SET UP .+2 ADR
10933  043600' 005724                   TST     (R4)+
10934  043602' 010423           10$:    MOV     R4,(R3)+                ;STORE .+2 ADR
10935  043604' 005023                   CLR     (R3)+                   ;STORE "HALT" INST CODE
10936  043606' 062704  000004           ADD     #4,R4                   ;INCR .+2 ADR
10937  043612' 020327  001000           CMP     R3,#1000                ;JUST DO 774 & 776?
10938  043616' 001371                   BNE     10$                     ;Y,N-10$
10939
10940                                           ;TAILOR & STORE EXEC INTERFACE ADR'S
10941
10942  043620' 010700                   MOV     PC,R0                   ;SET UP LOC 0 MPG ADR
10943  043622' 062700  134156           ADD     #CLOCZ-.,R0
10944  043626' 160200                   SUB     R2,R0                   ;SUB MPG'S POSITION ADJ FACTOR
10945  043630' 012701  000007           MOV     #CGBLTE-CGBLTS/2,R1     ;GET # OF WORDS IN GLOBAL TBL
10946  043634' 010702                   MOV     PC,R2                   ;SET UP GLOBAL TBL END ADR
10947  043636' 062702  001002           ADD     #CGBLTE-.,R2
10948  043642' 014203           CREL1:  MOV     -(R2),R3                ;GET RELATIVE ADR
10949  043644' 060703                   ADD     PC,R3                   ;MAKE IT ABSOLUTE
10950  043646' 010340           CHBASE: MOV     R3,-(R0)                ;STORE ABS ADR IN EXEC
10951  043650' 005301                   DEC     R1                      ;DECR WORD COUNT
10952  043652' 001373                   BNE     CREL1                   ;DONE ALL? (Y,N-CREL1)
10953
10954                                           ;TAILOR AND ISSUE MPG'S TITLE MSG
10955
10956  043654' 010700                   MOV     PC,R0                   ;SET UP RESTART POINT # 1 ADR
10957  043656' 062700  134212           ADD     #RST1-.,R0
10958  043662' 004567  175234           JSR     R5,BINASC               ;CONVERT IT TO ASCII & PUT IN MSG
10959  043666' 001167                   .WORD   CHKR1A-.
10960  043670' 010700                   MOV     PC,R0                   ;SET UP RESTART # 2 ADR
10961  043672' 062700  134232           ADD     #RST2-.,R0
10962  043676' 004567  175220           JSR     R5,BINASC               ;CONVERT IT ALSO
10963  043702' 001171                   .WORD   CHKR2A-.
10964  043704' 010700                   MOV     PC,R0                   ;GET MPG'S BASE ADR
```

```
10965  043706' 062700 134072            ADD     #CLOCZ-.,R0
10966  043712' 004567 175204            JSR     R5,BINASC               ;CONVERT IT TO ASCII &
10967  043716' 001212                   .WORD   CBASAD-.                ;PUT IN THE MSG
10968  043720' 004567 142300            JSR     R5,CTYPE                ;ISSUE MPG TITLE MSG
10969  043724' 000400                   .WORD   400
10970  043726' 001074                   .WORD   CTITLE-.
10971
10972                                    ;ADJUST COMPILER TABLES FOR RELOCATION
10973
10974  043730' 010703            LRELOC: MOV     PC,R3                   ;TABLE END ADDRESS
10975  043732' 062703 001066             ADD     #LMCRTE-.,R3            ;TO R3
10976  043736' 010700                    MOV     PC,R0                   ;TABLE START ADDRESS
10977  043740' 062700 000704             ADD     #LMCRTB-.,R0            ;TO R0
10978  043744' 010701            LRLLP1: MOV     PC,R1
10979  043746' 062001            LBA:    ADD     (R0)+,R1                ;CALC TABLE ADDRESS
10980  043750' 010702                    MOV     PC,R2
10981  043752' 062002            LBB:    ADD     (R0)+,R2                ;CALC SUBR ADDR
10982             001                    .IF DF MM
10983                                    MOV     PC,R3
10984                                    ADD     #PG6BA-.,R3             ;SET R3 = PAGE 6 BASE ADR
10985                                    SUB     R3,R2                   ;SUB FROM CALC ADDR
10986                                    ADD     #P6CONS,R2              ;ADD IN PAGE 6 CONSTANT
10987                                    MOV     PC,R3
10988                                    ADD     #LMCRTE-.,R3            ;PUT R3 BACK AS IT WAS
10989             000                    .ENDC
10990  043754' 010211                    MOV     R2,(R1)                 ;UPDATE COMPILER TABLE
10991  043756' 020003                    CMP     R0,R3
10992  043760' 001371                    BNE     LRLLP1                  ;REPEAT UNTIL COMPLETE
10993             001                    .IF NDF MM
10994  043762' 010703                    MOV     PC,R3
10995  043764' 062703 164222             ADD     #LCTBST+4-.,R3
10996  043770' 010700                    MOV     PC,R0
10997  043772' 062700 164046             ADD     #LCSTBL+4-.,R0
10998  043776' 010701                    MOV     PC,R1
10999  044000' 062701 175440             ADD     #CBUF00-.,R1            ;UPDATE BUFFER ADDRESSES
11000  044004' 010110                    MOV     R1,(R0)                 ;UPDATE BUFFER ADDRESSES
11001  044006' 062700 000006             ADD     #6,R0                   ;IN DIRECT REFERENCE
11002  044012' 010110            LRLLP2: MOV     R1,(R0)                 ;SYMBOL TABLE
11003  044014' 062700 000006             ADD     #6,R0
11004  044020' 062701 000020             ADD     #16.,R1
11005  044024' 020003                    CMP     R0,R3
11006  044026' 001371                    BNE     LRLLP2
11007  044030' 010703                    MOV     PC,R3
11008  044032' 062703 164366             ADD     #LXTBL+4-.,R3
11009  044036' 010701                    MOV     PC,R1
11010  044040' 062701 176000             ADD     #LCOM0-.,R1
11011  044044' 010110            LRLLP3: MOV     R1,(R0)                 ;UPDATE COMMON ADDRESSES
11012  044046' 062700 000006             ADD     #6,R0                   ;IN DIRECT REFERENCE
11013  044052' 062701 000002             ADD     #2,R1                   ;SYMBOL TABLE
11014  044056' 020003                    CMP     R0,R3
11015  044060' 001371                    BNE     LRLLP3
11016             000                    .ENDC
11017  044062' 010702                    MOV     PC,R2
11018  044064' 062702 163750             ADD     #LCSTBL-.,R2            ;CALCULATE
11019  044070' 010267 163252             MOV     R2,LDRTBS               ;DIR  REF TBL START
11020  044074' 010702                    MOV     PC,R2
```

```
11021  044076' 062702  164316          ADD     #LXTBL-.,R2             ;CALCULATE
11022  044102' 010267  163242          MOV     R2,LXRTBS               ;INDEX REF TBL START
11023  044106' 010702                  MOV     PC,R2
11024  044110' 062702  164510          ADD     #LINDTB-.,R2            ;CALCULATE
11025  044114' 010267  163232          MOV     R2,LIRTBS               ;INDIRECT REF TBL START
11026  044120' 010702                  MOV     PC,R2
11027  044122' 062702  164512          ADD     #LXINDT-.,R2            ;CALCULATE
11028  044126' 010267  163222          MOV     R2,LIITBS               ;INDEXED INDIRECT TBL START
11029  044132' 010702                  MOV     PC,R2
11030  044134' 062702  164514          ADD     #LSMEND-.,R2            ;CALCULATE
11031  044140' 010267  163212          MOV     R2,LSYMTE               ;SYMBOL TABLE END
11032  044144' 010702                  MOV     PC,R2                   ;MAKE BIT PACK TABLE
11033  044146' 062702  161336          ADD     #LBPAND-.,R2
11034  044152' 010267  161634          MOV     R2,LBPRT1
11035  044156' 010267  161634          MOV     R2,LBPRT2
11036  044162' 010702                  MOV     PC,R2                   ;POSITION INDEPENDANT
11037  044164' 062702  161422          ADD     #LBPTRU-.,R2
11038  044170' 010267  161626          MOV     R2,LBPRT3
11039  044174' 010267  161626          MOV     R2,LBPRT4
11040             001                   .IF DF MM
11041                                   .IFF
11042  044200' 010700                  MOV     PC,R0
11043  044202' 062700  175712          ADD     #LCRETN-.,R0            ;CALC POS IND ADDR
11044                                   .IFT
11045                                   MOV     #LCRETN-PG6BA+P6CONS,R0 ;CALC POS IND ADR
11046             000                   .ENDC
11047  044206' 010067  174104          MOV     R0,LPOCR1               ;FOR CR LF
11048  044212' 010067  174114          MOV     R0,LPOCR2
11049  044216' 010067  174354          MOV     R0,LPBCR1
11050             001                   .IF DF MM
11051                                   .IFF
11052  044222' 010700                  MOV     PC,R0                   ;CALC POS IND ADDR
11053  044224' 062700  170036          ADD     #LINBUF-.,R0            ;FOR LINBUF
11054                                   .IFT
11055                                   MOV     #LINBUF-PG6BA+P6CONS,R0 ;CALC POS IND ADR FOR LINBUF
11056             000                   .ENDC
11057  044230' 010067  173776          MOV     R0,LPOADR
11058  044234' 010067  174266          MOV     R0,LPBADR
11059             001                   .IF DF MM
11060                                   .IFF
11061  044240' 010700                  MOV     PC,R0                   ;MAKE PRINT PROG # S/R
11062  044242' 062700  173552          ADD     #LPN-.,R0               ;POSITION INDEPENDENT
11063  044246' 010067  173534          MOV     R0,LPNADR
11064  044252' 010700                  MOV     PC,R0                   ;MAKE  VERIFY ROUTINE
11065  044254' 062700  173116          ADD     #LVEHDR-.,R0            ;POSITION INDEPENDANT
11066  044260' 010067  172676          MOV     R0,LVECL1
11067  044264' 010700                  MOV     PC,R0
11068  044266' 062700  173170          ADD     #LVEMSG-.,R0
11069  044272' 010067  172730          MOV     R0,LVECL2
11070                                   .IFT
11071                                   MOV     #LPN-PG6BA+P6CONS,LPNADR ;MAKE PRT PRG # S/R POS IND
11072                                   MOV     #LVEHDR-PG6BA+P6CONS,LVECL1 ;MAKE VERIFY S/R
11073                                   MOV     #LVEMSG-PG6BA+P6CONS,LVECL2 ;POS IND
11074                                   MOV     PC,R0                   ;SET UP ABS FREE & MIDL ADR'S
11075                                   ADD     #FREEAD-.,R0            ;FOR MEM MGMNT VERSION
11076                                   MOV     R0,LFREE
```

```
11077                                          MOV     PC,R0
11078                                          ADD     #MIDLAD-.,R0
11079                                          MOV     R0,LMIDL
11080                  000                     .ENDC
11081                  001                     .IF NDF MIMIC
11082
11083                                          ;READ IN THE 'VALID DEVICES TABLE' FILE
11084
11085  044276' 010700                 LDVDTB: MOV     PC,R0           ;GET ADR OF TABLE'S AREA
11086  044300' 062700  176266                 ADD     #VDFLSZ-.,R0
11087  044304' 010067  000004                 MOV     R0,10$          ;STORE MEM ADR
11088  044310' 004577  133536                 JSR     R5,@LOADVR      ;GO LOAD THE FILE
11089  044314' 000000                 10$:    .WORD   XXXX            ;  MEM ADR
11090  044316' 001000                         .WORD   512.            ;  MAX BYTE COUNT
11091  044320' 000024                         .WORD   LDVDER-.        ;  DEV ERR ADR
11092  044322' 000022                         .WORD   LDVDER-.        ;  INSUF MEM ERR ADR
11093  044324' 000002                         .WORD   20$-.           ;  NORMAL RET ADR
11094  044326' 010700                 20$:    MOV     PC,R0           ;GET ADR OF 1ST WD IN TBL
11095  044330' 062700  176236                 ADD     #VDFLSZ-.,R0
11096  044334' 061000                         ADD     (R0),R0         ;COMPUTE END OF TBL ADR
11097                  002                     .IF DF MM
11098                                          BIT     #77,R0          ;AT A 32 WORD BNDRY?
11099                                          BEQ     25$             ;N,Y-25$
11100                                          ADD     #100,R0         ;BUMP UP PAST NXT BNDRY
11101                                  25$:    MOV     #6,R1           ;SET UP LOOP CNT
11102                                  35$:    CLC                     ;SHIFT ADR TO 32 WD BASE
11103                                          ROR     R0
11104                                          DEC     R1
11105                                          BNE     35$
11106                  001                     .ENDC
11107  044336' 010067  135112                 MOV     R0,CPAREA       ;STORE ADR AS START OF USR PROG AREA
11108  044342' 000406                         BR      HKVDTB          ;GO TO TBL HOUSEKEEPING
11109
11110  044344' 004567  141654         LDVDER: JSR     R5,CTYPE        ;ISSUE FILE LOAD ERROR MSG
11111  044350' 000602                         .WORD   602
11112  044352' 000565                         .WORD   LDVDEM-.
11113  044354' 000000                         HALT                    ;HALT FOR OPERATOR ACTION
11114  044356' 000747                         BR      LDVDTB          ;GO TRY LOAD AGAIN
11115                  000                     .ENDC
11116
11117                                          ;HOUSEKEEP SAVE/FETCH DEVICE BITS IN
11118                                          ;THE VALID DEVICES TABLE
11119
11120  044360' 017700  133440         HKVDTB: MOV     @CSAVEI,R0      ;GET LOAD DEV'S MDL CODE
11121  044364' 010701                         MOV     PC,R1           ;GET ADR OF MDL NAME TBL
11122  044366' 062701  176210                 ADD     #CMDLNM+4-.,R1
11123  044372' 011102                 10$:    MOV     (R1),R2         ;GET MDL CODE FROM TBL
11124  044374' 100414                         BMI     CKMFPS          ;END OF THE TBL? (N,Y-CKMFPS)
11125  044376' 042702  000360                 BIC     #360,R2         ;RESET DIFFERENTIATOR BITS
11126  044402' 020002                         CMP     R0,R2           ;THIS THE LOAD DEV'S ENTRY?
11127  044404' 001405                         BEQ     20$             ;N,Y-20$
11128  044406' 032711  000010                 BIT     #10,(R1)        ;PERMANENT SAV/FTH DEV?
11129  044412' 001002                         BNE     20$             ;N,Y-20$
11130  044414' 042711  000006                 BIC     #6,(R1)         ;CLEAR SAVE & FETCH SUPPORT BITS
11131  044420' 062701  000006         20$:    ADD     #6,R1           ;POINT AT NXT TBL ENTRY
11132  044424' 000762                         BR      10$             ;GO CK IT
```

```
11133
11134                                   ;CHECK IF CPU NEEDS MFPS/MTPS INSTRUCTIONS AND
11135                                   ;IT IT DOES, CHECK IF IT IS AN LSI-11.
11136
11137  044426' 010700                  CKMFPS: MOV     PC,R0                   ;SET UP ADR OF HOUSEKEEPING'S
11138  044430' 062700  000164                  ADD     #HSKTRP-.,R0            ;TRAP ROUT
11139  044434' 010037  000004                  MOV     R0,@#4                  ;STORE IN CPU ERRORS TRAP VECTOR
11140  044440' 010037  000010                  MOV     R0,@#10                 ;STORE IN RESERVED INST TRAP VECT
11141  044444' 106700                          MFPS    R0                      ;ISSUE INST TO CK CPU TYPE
11142  044446' 052767  000002  167540          BIS     #USMTPS,CSYSFW          ;DID IT - SET FLG TO INDICATE CPU TYPE
11143  044454' 000401                          BR      30$                     ;GO TO LSI-11 CHECK
11144  044456' 000407                          BR      CKRSET                  ;IT TRAPPED - BYPASS LSI-11 TEST
11145  044460' 010700                  30$:    MOV     PC,R0                   ;GET AN EVEN ADR
11146  044462' 005200                          INC     R0                      ;MAKE IT ODD
11147  044464' 005710                          TST     (R0)                    ;DO WORD INST TO AN ODD ADR
11148                                                                          ;LSI-11 WILL NOT TRAP
11149  044466' 052767  000004  167520          BIS     #LSI11,CSYSFW           ;NO TRAP - SET LSI-11 FLAG
11150  044474' 000402                          BR      CKRST2                  ;BYPASS VECTOR RESET
11151  044476' 005037  000102          CKRSET: CLR     @#102                   ;RESET LSI-11 CLOCK VECTOR
11152  044502' 012737  000012  000010  CKRST2: MOV     #12,@#10                ;RESTORE RESERVED INST TRAP VECT
11153
11154
11155                                   ;TEST IF THE CPU HAS A CACHE MEMORY BY
11156                                   ;ATTEMPTING TO ACCESS THE HIT/MISS REGISTER.
11157
11158  044510' 005737  177752          CK4CHE: TST     @#177752                ;ACCESS THE HIT/MISS REG
11159  044514' 012767  000002  167742          MOV     #2,CTOADJ               ;ITS THERE; ADJ T/O VALUE
11160  044522' 000401                          BR      31$                     ;BYPASS TRAP BR
11161  044524' 000406                          BR      CK4A70                  ;NOT THERE - GO TO NEXT TEST
11162  044526' 012767  002126  172234  31$:    MOV     #2126,LDLLP1+2          ;ADJ DELAY INST TIMING VALUE
11163  044534' 052767  000200  167452          BIS     #CACHE,CSYSFW           ;SET CACHE MEM PRESENT FLAG
11164
11165
11166                                   ;TEST IF THE CPU IS AN 11/70 BY ATTEMPTING TO SET
11167                                   ;THE 22 BIT ADDRESSING BIT IN MMMR3.
11168
11169  044542' 012737  000020  172516  CK4A70: MOV     #20,@#MMMR3             ;TRY TO SET 22 BIT ADR BIT
11170  044550' 032737  000020  172516          BIT     #20,@#MMMR3             ;DID BIT SET?
11171  044556' 001001                          BNE     32$                     ;N,Y-32$
11172  044560' 000405                          BR      33$                     ;GO TO NEXT TEST
11173  044562' 052767  000010  167424  32$:    BIS     #CPU70,CSYSFW           ;SET THE CPU = 11/70 FLAG
11174  044570' 005037  172516                  CLR     @#MMMR3                 ;RESET 22 BIT ADR BIT
11175             001                          .IF DF MM
11176                                          .IFF
11177  044574' 012737  000006  000004  33$:    MOV     #6,@#4                  ;RESTORE TRAP VECTOR
11178
11179
11180                                          .IFT
11181                                   33$:
11182
11183
11184                                   ;TEST IF CPU IS AN 11/40 BY ATTEMPTING TO ACCESS
11185                                   ;MMMR3. IF WE CAN, ITS AN 11/45 OR HIGHER.
11186
11187                                   CK4A40: BIS     #CPU40,CSYSFW           ;INITIALIZE TO AN 11/40
11188                                           TST     @#MMMR3                 ;TRY TO ACCESS MMMR3
```

```
11189                                   BIC     #CPU40,CSYSFW           ;DID IT - RESET 11/40 FLG
11190                                   NOP                             ;FILL INST
11191
11192                                   ;INITIALIZE MEMORY MANAGEMENT REGISTERS AND
11193                                   ;ENABLE ITS OPERATION.
11194
11195                           INITMM: MOV     #KPAR0,R0               ;SET UP ADR'S OF KERNEL MODE REGS
11196                                   MOV     #KPDR0,R1
11197                                   MOV     #8.,R2                  ;SET UP REGISTER COUNT
11198                                   CLR     R3                      ;SET MEM BASE ADR TO 0
11199                           35$:    MOV     R3,(R0)+                ;STORE PAGE'S MEM BASE ADR
11200                                   MOV     #PDRCON,(R1)+           ;STORE DESCRIPTOR REG VALUE
11201                                   ADD     #200,R3                 ;POINT TO NEXT MEM PAGE
11202                                   DEC     R2                      ;DONE ALL REGISTERS?
11203                                   BNE     35$                     ;Y,N-35$
11204                                   MOV     #IOPAGE,-(R0)           ;SET PAR 7 TO THE I/O PAGE
11205                                   BIT     #CPU40,@SYSFW           ;THIS AN 11/40?
11206                                   BNE     40$                     ;N,Y-40$
11207                                   CLR     @#MMMR3                 ;HOUSEKEEP MMMR3 REG
11208                           40$:    MOV     #1001,@#MMMR0           ;ENABLE MEM MGMNT
11209
11210                                   ;NOW RUNNING UNDER MEMORY MANAGEMENT
11211
11212                                   ;CHECK IF THE UNIBUS MAP AND 22 BIT ADDRESSING
11213                                   ;ARE AVAILABLE ON THIS CPU. IF THEY ARE, ASK IF
11214                                   ;THEY ARE TO BE USED.
11215
11216                           CKUMAP: BIT     #CPU40,CSYSFW           ;THIS AN 11/40?
11217                                   BNE     MMSZM                   ;N,Y-MMSZM
11218                                   MOV     #MMMR3,R3               ;SET UP MM REG ADR
11219                                   MOV     #60,(R3)                ;ENABLE THE MAP & 22 BIT ADDRESSING
11220                                   BIT     #40,(R3)                ;DID THE UNIBUS MAP BIT SET?
11221                                   BEQ     CK22AD                  ;Y,N-CK22AD
11222                           42$:    JSR     R5,CTYPE                ;ASK IF THEY WANT US TO USE IT
11223                                   .WORD   420
11224                                   .WORD   UBMPMG-.
11225                                   .WORD   26.
11226                                   JSR     PC,CKRPLY               ;CHECK THEIR REPLY
11227                                   BR      42$                     ;BR IF INV REPLY
11228                                   BR      CK22AD                  ;BR IF REPLY IS 'N'
11229                                   BIS     #UNIMAP,CSYSFW          ;SET THE "USE UNIBUS MAP" FLAG
11230                                   BIS     #40,MR3VAL              ;SET ITS BIT IN WD FOR MMMR3 SET UP
11231                           CK22AD: BIT     #20,(R3)                ;DID THE 22 BIT ADR BIT SET?
11232                                   BEQ     CKUMCL                  ;Y,N-CKUMCL
11233                           44$:    JSR     R5,CTYPE                ;ASK IF THEY WANT TO USE
11234                                   .WORD   420                     ;22 BIT ADDRESSING
11235                                   .WORD   U22BAD-.
11236                                   .WORD   27.
11237                                   JSR     PC,CKRPLY               ;CHECK THEIR REPLY
11238                                   BR      44$                     ;BR IF INV REPLY
11239                                   BR      CKUMCL                  ;BR IF REPLY IS 'N'
11240                                   BIS     #BITS22,CSYSFW          ;SET THE "USE 22 BITS" FLAG
11241                                   BIS     #20,MR3VAL              ;SET ITS BIT IN WD FOR MMMR3 SET UP
11242                                   MOV     #177600,IOPAR7          ;SET 22 BIT I/O PAGE PAR
11243                           CKUMCL: BIC     #40,(R3)                ;RESET THE UNIBUS MAP FOR NOW
11244
```

```
11245                                   ;SIZE MEMORY ABOVE THE 16K MEMORY BOUNDARY
11246
11247                           MMSZM:  MOV     #P6CONS,R0              ;SET UP ADR THAT SELECTS PAR 6
11248                                   MOV     #1000,@#KPAR6           ;INITIALIZE PAR 6 TO 16K WD BNDRY
11249                           50$:    BIT     #BIT22,CSYSFW           ;USING 22 BIT ADDRESSING?
11250                                   BEQ     60$                     ;Y,N-60$
11251                                   CMP     #170000,@#KPAR6         ;END OF 22 BIT MEMORY?
11252                                   BEQ     80$                     ;N,Y-80$
11253                                   BR      70$                     ;CONTINUE MEM TEST
11254                           60$:    CMP     #7600,@#KPAR6           ;END OF 18 BIT MEMORY?
11255                                   BEQ     80$                     ;N,Y-80$
11256                           70$:    TST     (R0)                    ;TEST CURRENT MEM LOCATION
11257                                   ADD     #40,@#KPAR6             ;INCR TO NEXT 1K OF MEM
11258                                   BR      50$                     ;GO TEST NEXT LOC ADR
11259                           80$:    MOV     @#KPAR6,CRMEND          ;STORE LAST K OF MEM ADR
11260                                   DEC     CRMEND
11261                                   MOV     #6,@#4                  ;RESTORE CPU ERROR TRAP VECT
11262                                   CLR     @#MMR0                  ;DISABLE MEM MGMNT
11263
11264                                   ;MEMORY MANAGEMENT IS NOW DISABLED
11265
11266                                   ;INITIALIZE SHARED PAGE PAR & PDR VALUES
11267
11268                           IPG6VL: MOV     PC,R1                   ;GET START ADR OF SHARED PAGE
11269                                   ADD     #PG6BA-.,R1
11270                                   MOV     PC,R0                   ;GET ITS END ADR
11271                                   ADD     #PG6END-.,R0
11272                                   SUB     R1,R0                   ;GET ITS SIZE IN BYTES
11273                                   BIT     #77,R0                  ;END ON A 32 WD BNDRY?
11274                                   BEQ     82$                     ;N,Y-82$
11275                                   ADD     #100,R0                 ;POINT IT PAST NXT BNDRY
11276                           82$:    MOV     #6,R2                   ;CONVERT ADR TO A 32 WD ADR
11277                           84$:    ASR     R0                      ;AND BYTE CNT TO A 32 WD
11278                                   ASR     R1                      ;BLK CNT
11279                                   DEC     R2
11280                                   BNE     84$
11281                                   MOV     R1,SPPAR6               ;STORE SHARED PAGE PAR VALUE
11282                                   DEC     R0                      ;ADJ BLK CNT
11283                                   SWAB    R0                      ;GET PLF BITS IN HIGH BYTE
11284                                   BIS     #6,R0                   ;SET IN ACF BITS
11285                                   MOV     R0,SPPDR6               ;STORE SHARED PAGE PDR VALUE
11286
11287                                   ;INITIALIZE TRAP VECTORS
11288
11289                           ITRPVC: MOV     PC,R0                   ;GET TRAP INST ROUT ABS ADR
11290                                   ADD     #TRP34-.,R0
11291                                   MOV     R0,@#34                 ;STORE IT AT ITS VECTOR
11292                                   MOV     PC,R0                   ;SET UP ADR OF ROUT FOR
11293                                   ADD     #TRP04-.,R0             ;CPU ERROR TRAP
11294                                   MOV     R0,@#4                  ;STORE IT
11295                                   MOV     PC,R0                   ;SET UP ADR OF ROUT FOR
11296                                   ADD     #TRP10-.,R0             ;RESERVED INST TRAP
11297                                   MOV     R0,@#10                 ;STORE IT
11298                                   MOV     PC,R0                   ;SET UP ADR OF ROUT FOR
11299                                   ADD     #TRP114-.,R0            ;MEMORY ERRORS
11300                                   MOV     R0,@#114                ;STORE IT
```

```
11301                                   MOV     PC,R0                   ;SET UP ADR OF ROUT FOR
11302                                   ADD     #TRP250-.,R0            ;MEM MGMNT TRAP
11303                                   MOV     R0,@#250                ;STORE IT
11304           000                     .ENDC
11305           001                     .IF DF MIMIC
11306
11307                                   ;CHECK IF DEV ROUT IS INCLUDED & ADJUST MPG
11308
11309                           CK4DVR: MOV     PC,R0                   ;SET UP MPG END ADR
11310                                   ADD     #MPGEND-.,R0
11311                                   MOV     R0,R3                   ;SAVE IT
11312           002                     .IF DF MM
11313                                   MOV     #6,R1                   ;SET UP LOOP CNT
11314                           90$:    CLC                             ;EXTRACT HIGH 16 BITS OF THE ADR
11315                                   ROR     R0
11316                                   DEC     R1
11317                                   BNE     90$
11318           001                     .ENDC
11319                                   MOV     R0,CPAREA               ;SAVE NEW USR PROG START ADR
11320                                   CMP     LIMIT+2,R3              ;IS A DEV ROUT LINKED IN?
11321                                   BLOS    HSKEND                  ;Y,N-HSKEND
11322                                   JSR     R5,BINASC               ;PLACE DVR ST ADR IN MSG
11323                                   .WORD   CDVRBG-.
11324                                   MOV     PC,R0                   ;SET UP SLOT # 1 ADR
11325                                   ADD     #CRTBL-.,R0
11326                                   JSR     R5,BINASC               ;PLACE SLOT # 1 ADR IN MSG
11327                                   .WORD   CSLT1A-.
11328           002                     .IF DF MM
11329                                   MOV     PLNGTH(R3),R0           ;GET DEV ROUT SIZE
11330                                   BIT     #77,R0                  ;END ON A 32 WD BNDRY?
11331                                   BEQ     100$                    ;N,Y-100$
11332                                   ADD     #100,R0                 ;POINT PAST NXT BNDRY
11333                           100$:   MOV     #6,R1                   ;SET UP LOOP CNT
11334                           105$:   ASR     R0                      ;GET # OF 32 WD BLKS
11335                                   DEC     R1
11336                                   BNE     105$
11337                                   MOV     R0,CRTBL+34.            ;STORE IN CNTRL ROUT TBL EXT
11338           001                     .ENDC
11339                                   JSR     R5,CTYPE                ;TYPE THE "WRM" MSG
11340                                   .WORD   400
11341                                   .WORD   CWRMSG-.
11342           002                     .IF DF MM
11343                                   BIT     #UNIMAP,CSYSFW          ;USING THE UNIBUS MAP?
11344                                   BNE     115$                    ;N,Y-115$
11345                                   CLR     PUBMAP(R3)              ;RESET ITS MAP INFO
11346           001                     .ENDC
11347                           115$:   ADD     #PTLGTH+DEVIOB,R3       ;POINT AT DVR ADDRESSES
11348                                   MOV     PC,R0                   ;SET UP TABLE ADR
11349                                   ADD     #CLDTBL-.,R0
11350                           CHKLOP: MOV     (R0)+,R2                ;GET REL OR VIRT MPG POINT ADR
11351                                   BEQ     HSKEND                  ;END OF TBL? (N,Y-HSKEND)
11352           002                     .IF DF MM
11353                                   BMI     CHKVAD                  ;PAGE 6 ADR? (N,Y-CHKVAD)
11354           001                     .ENDC
11355                                   ADD     PC,R2                   ;MAKE IT ABS
11356                           CHKBAS: SUB     #CHKBAS-CLDBAS,R2       ;ADJ FOR DIFF PC ADRS
```

```
11357                                   CHKVAD: MOV     R2,(R3)+                ;STORE IT IN DVR
11358                                           BR      CHKLOP                  ;GO GET NXT ADR
11359
11360                    000                    .ENDC
11361  044602' 012745  000240          HSKEND: MOV     #240,-(R5)              ;SET CALLING JSR TO NOP'S
11362  044606' 012745  000240                  MOV     #240,-(R5)
11363  044612' 000205                          RTS     R5                      ;EXIT IN-LINE
11364
11365
11366                                           ;HOUSEKEEPING'S TRAP ROUTINE
11367
11368  044614' 062716  000010          HSKTRP: ADD     #8.,(SP)                ;BYPASS NEXT 4 WDS OF PROG
11369  044620' 000002                          RTI                             ;EXIT FROM THIS TRAP
11370
11371    .   .
11372                    001                    .IF DF MM
11373                                           ;GET REPLY FOR HSKP MESSAGE AND CHECK IT.
11374                                           ;Y, N, <CR>, & <LF> ARE ONLY VALID REPLIES.
11375                                           ;
11376                                           ;       JSR     PC,CKRPLY       S/R CALL
11377                                           ;       BR      LABEL           EXECUTED IF INVALID REPLY
11378                                           ;       BR      LABEL           EXECUTED IF A 'N' REPLY
11379                                           ;
11380                                           ;DESTROYS R0,R1,R2
11381
11382                                   CKRPLY: JSR     R5,@CTRD                ;ISSUE KEYBOARD READ
11383                                           .WORD   CCMDRD-.
11384                                           .WORD   30.
11385                                           JSR     R5,CSCNSP               ;SCAN READ DATA FOR THE REPLY
11386                                           BIT     #CSCFND,R2              ;ONLY <CR> OR <LF> ENTRY?
11387                                           BEQ     10$                     ;N,Y-10$
11388                                           CMPB    #'N,@CDSTAD             ;IS REPLY AN 'N'?
11389                                           BEQ     20$                     ;N,Y-20$
11390                                           CMPB    #'Y,@CDSTAD             ;IS REPLY A 'Y'?
11391                                           BNE     30$                     ;Y,N-30$
11392                                   10$:    ADD     #2,(SP)                 ;BYPASS 2ND BR INST
11393                                   20$:    ADD     #2,(SP)                 ;BYPASS 1ST BR INST
11394                                   30$:    RTS     PC                      ;EXIT
11395                    000                    .ENDC
11396
11397
11398  044622' 147404                  CGBLTS: .WORD   UFWSET-CHBASE           ;GLOBAL ADDRESSES TABLE
11399  044624' 147422                          .WORD   UFWCLR-CHBASE
11400  044626' 147440                          .WORD   UFWTST-CHBASE
11401  044630' 175254                          .WORD   BINASC-CHBASE
11402  044632' 175316                          .WORD   BTASLB-CHBASE
11403  044634' 170346                          .WORD   CSYSFW-CHBASE
11404  044636' 134360                          .WORD   USRINT-CHBASE
11405          044640'                  CGBLTE  =.
11406
11407
11408  044640' 000000  000000          LIMIT:  .LIMIT                          ;MPG'S START & END ADR'S
11409
11410                                           ;
11411                                           ;       COMPILER MACHINE CODE RELOCATION TABLE
11412                                           ;
```

```
11413  044644' 165054  171750          LMCRTB: .WORD   LRT1-LBA,LADDM-LBB
11414  044650' 165070  172046                  .WORD   LRT2-LBA,LSUBM-LBB
11415  044654' 165442  174050                  .WORD   LRT3-LBA,LPRNG-LBB
11416  044660' 165454  174050                  .WORD   LRT4-LBA,LPRNG-LBB
11417  044664' 165466  174154                  .WORD   LRT5-LBA,LPRNO-LBB
11418  044670' 165500  174142                  .WORD   LRT6-LBA,LPRND-LBB
11419  044674' 165512  174430                  .WORD   LRT7-LBA,LPRNB-LBB
11420  044700' 165524  174050                  .WORD   LRT8-LBA,LPRNG-LBB
11421  044704' 165536  174154                  .WORD   LRT9-LBA,LPRNO-LBB
11422  044710' 165550  174142                  .WORD   LRT10-LBA,LPRND-LBB
11423  044714' 165562  174430                  .WORD   LRT11-LBA,LPRNB-LBB
11424  044720' 165574  172244                  .WORD   LRT12-LBA,LFILR-LBB
11425  044724' 165606  172322                  .WORD   LRT13-LBA,LFILG-LBB
11426  044730' 165620  172200                  .WORD   LRT14-LBA,LFILV-LBB
11427  044734' 165634  172244                  .WORD   LRT15-LBA,LFILR-LBB
11428  044740' 165646  172322                  .WORD   LRT16-LBA,LFILG-LBB
11429  044744' 165660  172200                  .WORD   LRT17-LBA,LFILV-LBB
11430  044750' 165724  172760                  .WORD   LRT18-LBA,LROTR-LBB
11431  044754' 165736  172760                  .WORD   LRT19-LBA,LROTR-LBB
11432  044760' 165750  173012                  .WORD   LRT20-LBA,LDLAY-LBB
11433  044764' 165430  174050                  .WORD   LRT21-LBA,LPRNG-LBB
11434  044770' 165674  172354                  .WORD   LRT22-LBA,LFILP-LBB
11435  044774' 165710  172354                  .WORD   LRT23-LBA,LFILP-LBB
11436  045000' 166140  173032                  .WORD   LRT27-LBA,LVERFY-LBB
11437  045004' 166154  173032                  .WORD   LRT28-LBA,LVERFY-LBB
11438  045010' 166234  171244                  .WORD   LRT29-LBA,CUPCR-LBB
11439  045014' 166314  171044                  .WORD   LRT30-LBA,CUPGEX-LBB
11440  045020' 000000          LMCRTE: .WORD   0                       ;END OF RELOCATION TABLE
11441
11442                                  .NLIST  BEX
11443             001                  .IF DF MM
11444                                  .IFF
11445  045022' 005015  052104  043515  CTITLE: .ASCII  <015><012>/DTMGA-C   M.P.G.   RST1: /
11446                                  .IFT
11447                                  CTITLE: .ASCII  <015><012>/DTMMA-B   M.P.G.   RST1: /
11448             000                  .ENDC
11449  045055'    130  054130  054130  CHKR1A: .ASCII  /XXXXXX; RST2: /
11450  045073'    130  054130  054130  CHKR2A: .ASCII  /XXXXXX/<015><012><015><012>
11451  045105'    115  043520  041040          .ASCII  /MPG BASE ADDRESS = /
11452  045130' 054130  054130  054130  CBASAD: .ASCIZ  /XXXXXX/
11453             001                  .IF DF MIMIC
11454                                  CWRMSG: .ASCII  <015><012>/DO: "WRM /
11455                                  CSLT1A: .ASCII  /XXXXXX,/
11456                                  CDVRBG: .ASCIZ  /XXXXXX*, THEN "FM"/
11457             000                  .ENDC
11458             001                  .IF NDF MIMIC
11459  045137'    126  042114  042040  LDVDEM: .ASCIZ  /VLD DEV TBL LD ERR, PRESS "CONT" TO RETRY/
11460             000                  .ENDC
11461             001                  .IF DF MM
11462                                  UBMPMG: .ASCII  <015><012>'?USE UNIBUS MAP (Y/N) / '
11463                                  U22BAD: .ASCII  <015><012>'?USE 22 BIT ADRS (Y/N) / '
11464             000                  .ENDC
11465                                  .LIST   BEX
11466
11467          045212'                 .EVEN
11468             001                  .IF DF MIMIC & MM
```

```
11469
11470                                   ALPT2=  .-CLOCZ
11471                                   ALFCT2= ALPT2&000077/2
11472           002                             .IF NE ALFCT2
11473                                   ALQUN2= 31.-ALFCT2
11474                                           .BLKW   ALQUN2
11475                                           .WORD   0
11476           001                             .ENDC
11477
11478           000                     .ENDC
11479        045212'                    MPGEND= .
11480        000001                             .END
```

```
ACTIVE= 100000          CBUF05  041560R 002   CERM03  006625R 002   CHBASE  043646R 002   CMIDLM  007513R 002
ADR18B  033462R 002     CBUF06  041600R 002   CERM04  006640R 002   CHKR1A  045055R 002   CMMEND  007466R 002
ASDEC0  042350R 002     CBUF07  041620R 002   CERM05  006645R 002   CHKR2A  045073R 002   CMMHDR  007416R 002
ASDEC1  042342R 002     CBUF08  041640R 002   CERM06  006663R 002   CHPINT= 004000          CMMMDL  007506R 002
AURPEP= 000010          CBUF09  041660R 002   CERM07  006671R 002   CINACT  002510R 002   CMMMSG  007451R 002
AUTORP= 000020          CBUF10  041700R 002   CERM08  006701R 002   CINTMG  010016R 002   CMMPNM  007476R 002
BCPTSF  033364R 002     CBUF11  041720R 002   CERM09  006715R 002   CINTPC  010034R 002   CMMSTA  007454R 002
BINASC  041122R 002     CBUF12  041740R 002   CERM10  006727R 002   CINTPN  010066R 002   CMODIF  002434R 002
BINASI  041126R 002     CBUF13  041760R 002   CERM11  006737R 002   CINTSP  010050R 002   CMPCKE  004272R 002
BINASB  041122R 002     CBUF14  042000R 002   CERM12  006751R 002   CIOALT= 000400          CMPNXP  004276R 002
BIT11 = 004000          CBUF15  042020R 002   CERM13  006760R 002   CIOBSY  035216R 002   CMPPFM  004404R 002
BIT12 = 010000          CCBD    006210R 002   CERM14  006765R 002   CIOSZM  010073R 002   CMPPPG  004310R 002
BIT6  = 000100          CCDB    006164R 002   CERM15  007003R 002   CISDNM  013472R 002   CMVNAM  004722R 002
BOOT    000006R 002     CCDBER  006200R 002   CERM16  007017R 002   CIVMSG  007243R 002   CMVNEX  004644R 002
BTASLB  041164R 002     CCFMTA  001750R 002   CERM17  007026R 002   CKBDID  042600R 002   CNCMND  000366R 002
BTASLZ  041172R 002     CCFMTB  001762R 002   CERM18  007035R 002   CKCLRG  035140R 002   CNCTU   000764R 002
BTPACK  025314R 002     CCFMTJ  001734R 002   CERM19  007041R 002   CKCMND  000530R 002   CNONID  042572R 002
BTUPAK  026030R 002     CCFMT1  001710R 002   CERM20  007046R 002   CKEX    035172R 002   CNSMSG  007247R 002
BUF     000010R 002     CCFMT2  002004R 002   CERM21  007051R 002   CKILEX  035214R 002   CNTRLO= 010000
BUMP    036114R 002     CCFMT3  002020R 002   CERM22  007056R 002   CKILL   002520R 002   CN20ER  002030R 002
CAAMSG  007335R 002     CCFMT4  001676R 002   CERM23  007067R 002   CKMFPS  044426R 002   CN20PJ  005666R 002
CACHE = 000200          CCFPBJ  002712R 002   CERM24  007132R 002   CKODAD  013344R 002   COMPIL  015030R 002
CADD    006042R 002     CCFPDE  001740R 002   CERM25  007141R 002   CKPDVP  035032R 002   COMRET  003342R 002
CADMSG  007533R 002     CCFRET  001756R 002   CERM26  007151R 002   CKRSET  044476R 002   COPER   002716R 002
CAIDLE= 001000          CCMDRD  001464R 002   CERM27  007163R 002   CKRST2  044502R 002   COPSW   002636R 002
CANS    007724R 002     CCONT   002476R 002   CERM30  007174R 002   CK4A70  044542R 002   COPSWN  002726R 002
CANSMG  007716R 002     CDCNT   001600R 002   CERM31  007207R 002   CK4CHE  044510R 002   CPAREA  001454R 002
CARDSP  006072R 002     CDDFP = 100000        CERM32  007235R 002   CLDBAS  012456R 002   CPATBL  001064R 002
CARITH  006062R 002     CDELET  002414R 002   CERRET  001602R 002   CLDER1  012476R 002   CPCMRF  007753R 002
CARIT1  006064R 002     CDFNAM  001644R 002   CESMSG  007733R 002   CLDER2  012502R 002   CPGCON  035370R 002
CARRET  006102R 002     CDISPL  003240R 002   CEXPRG  034632R 002   CLDER3  012506R 002   CPGCPE  034472R 002
CASAVE  005452R 002     CDISPN  007560R 002   CFCMND  000350R 002   CLDEVR  011642R 002   CPGCPT  034666R 002
CASCOM  005516R 002     CDLCEX  005124R 002   CFECNR  004652R 002   CLDEX   012464R 002   CPGDSP  034434R 002
CASER0  005636R 002     CDNAME  007564R 002   CFETCH  004570R 002   CLDID   042614R 002   CPGHSK  035404R 002
CASER1  005640R 002     CDREND  001616R 002   CFIER1  004052R 002   CLDISU  012376R 002   CPGNXT  034604R 002
CASER2  005650R 002     CDSHDR  007547R 002   CFIER2  004062R 002   CLDLOP  012452R 002   CPGRSV  035224R 002
CASEX   005632R 002     CDSTAD  001576R 002   CFIER3  004072R 002   CLDNRT  012254R 002   CPGTOE  034736R 002
CASE1J  005442R 002     CDVRST  012242R 002   CFIER4  004102R 002   CLDPDR  012026R 002   CPGTTE  034470R 002
CASFCM  005464R 002     CDVRSZ  012244R 002   CFIER5  004112R 002   CLDTBL  012522R 002   CPNASC  013610R 002
CASFTH  005436R 002     CECMSG  007314R 002   CFIEX   004250R 002   CLIST   035600R 002   CPNAS1  013624R 002
CASLST  005474R 002     CENDER  002376R 002   CFILL   003460R 002   CLISTI  000014R 002   CPNMSG  007365R 002
CASMSG  007331R 002     CENERA  002404R 002   CFIPAT  004122R 002   CLOCZ   000000R 002   CPNTR   001462R 002
CASSIG  005300R 002     CENERR  002402R 002   CFMEND  001450R 002   CLOSE   000016R 002   CPRT    000020R 002
CBASAD  045130R 002     CENTER  002040R 002   CFMMSG  007435R 002   CLRVEC  040714R 002   CPSVR5  035226R 002
CBASE1  001056R 002     CENTPA  002062R 002   CFMST   001446R 002   CLSJMP  002472R 002   CPU70 = 000010
CBASE2  001024R 002     CENTPB  002356R 002   CFTCHF= 040000        CLST    005042R 002   CRATBL  001074R 002
CBOC    006134R 002     CENTPC  002352R 002   CFTCHI  000012R 002   CMAP    004254R 002   CRDM    005716R 002
CBOOT   005270R 002     CEPNUM  007374R 002   CGBLTE= 044640R 002   CMCDVN  001620R 002   CRDMDT  007407R 002
CBUF00  041440R 002     CERDTA  007300R 002   CGBLTS  044622R 002   CMDLNM  042572R 002   CRDMG1  007377R 002
CBUF01  041460R 002     CERID   007360R 002   CGOENT  004614R 002   CMDTBL  001162R 002   CRDMG2  007406R 002
CBUF02  041500R 002     CERMSG  007265R 002   CGOREX  035524R 002   CMEMCK  013444R 002   CREATE  000022R 002
CBUF03  041520R 002     CERM01  006600R 002   CGORPT  035504R 002   CMGCNT  000436R 002   CREL1   043642R 002
CBUF04  041540R 002     CERM02  006607R 002   CGORUN  000470R 002   CMIDL   007523R 002   CRENPA  007607R 002
```

DTMGAC.P11 SYMBOL TABLE

```
CREPOR  003346R  002   CSUERR  013400R  002   DERPAD= 000036          FILPEX  036506R  002   LBPLP1  025374R  002
CRFLGW  001444R  002   CSUPTR  013330R  002   DEVBTA= 000056          FLLDMG  010154R  002   LBPLP2  025460R  002
CRMEND  001452R  002   CSVCOM  004522R  002   DEVDER= 000050          FLLDNM  010170R  002   LBPLP3  025554R  002
CRPFLG  035522R  002   CSYSFW  034214R  002   DEVDRA= 000024          FSTTME  016102R  002   LBPLP4  025676R  002
CRPTEP  035476R  002   CSZREQ  011674R  002   DEVETP= 000104          GET     000046R  002   LBPLP5  025722R  002
CRQADR  011172R  002   CTFLGW  000026R  002   DEVFWD= 000002          GETBYT  040766R  002   LBPMOR  025422R  002
CRQCTA  011610R  002   CTITLE  045022R  002   DEVIOB= 000046          GTNXTD= 001000          LBPNF   025472R  002
CRQDRD  007640R  002   CTOADJ  034464R  002   DEVIVA= 000026          HKVDTB  044360R  002   LBPPGL  025654R  002
CRQDRM  007625R  002   CTOCK   034466R  002   DEVIW1= 000004          HSKEND  044602R  002   LBPRT1  026012R  002
CRQER   011630R  002   CTPRIO= 000020          DEVIW2= 000006          HSKPEP= 000004          LBPRT2  026016R  002
CRQEX   011606R  002   CTRD    000030R  002   DEVIW3= 000010          HSKTRP  044614R  002   LBPRT3  026022R  002
CRQLNK  011340R  002   CTRDEX  000032R  002   DEVIW4= 000012          INVSTK  035526R  002   LBPRT4  026026R  002
CRQLPT  011332R  002   CTRDNE  000034R  002   DEVIW5= 000014          JCPGNV  033026R  002   LBPRWD  026004R  002
CRQNXM  011230R  002   CTRQBP= 100000          DEVIW6= 000016          JCPTSF  033052R  002   LBPSBO  025570R  002
CRSCMD  000754R  002   CTUDED= 020000          DEVIW7= 000020          JLPKER  017752R  002   LBPST1  025752R  002
CRTBL   034216R  002   CTUSWR  000036R  002   DEVIW8= 000022          LADD    035772R  002   LBPTG1  025406R  002
CRUDED= 040000         CTWR    000040R  002   DEVRPS= 000030          LADDCC  035766R  002   LBPTG2  025714R  002
CRUER1  003200R  002   CTYPE   006224R  002   DEVRSZ= 000000          LADDEX  036014R  002   LBPTG3  025740R  002
CRUER2  003210R  002   CUPCR   035216R  002   DEVSTP= 000102          LADDM   035722R  002   LBPTRU  025606R  002
CRUER3  003220R  002   CUPGER  034760R  002   DEVWPS= 000032          LADDML  035752R  002   LBPVTB  026010R  002
CRUER4  003230R  002   CUPGEX  035016R  002   DFLTMN  012732R  002   LADD1   022420R  002   LBRK1   022411R  002
CRUN    002760R  002   CURCTE  001456R  002   DHKPAD= 000034          LADD2   022425R  002   LBRK2   023240R  002
CR1ADR  001604R  002   CURPTA  001460R  002   DIDTAD  042570R  002   LASCNT  042410R  002   LBTEX   042316R  002
CR2ADR  001606R  002   CUSPGR= 002000          DISP    024470R  002   LASDEC  042352R  002   LBTNBR  026002R  002
CSACNR  004646R  002   CUSPRT  000042R  002   DISPCD  024476R  002   LASDON  023252R  002   LBTNSV  026006R  002
CSAINM  005000R  002   CVLCOM  012614R  002   DKILAD= 000040          LASERR  042522R  002   LBTPNT  025776R  002
CSANAS  004760R  002   CVLER1  013210R  002   DOERCK= 000400          LASER1  042524R  002   LBUAND  026336R  002
CSANNM  004770R  002   CVLER2  013220R  002   DOIOT = 000040          LASEX   042514R  002   LBUAST  026346R  002
CSANPN  005010R  002   CVLER3  013230R  002   DRTEND= 000116          LASLP0  042372R  002   LBUBKU  026222R  002
CSAVE   004512R  002   CVLER4  013244R  002   DRTLTH= 000116          LASLP1  042420R  002   LBUBPT  026324R  002
CSAVEI  000024R  002   CVLFIL  013154R  002   DTOEAD= 000044          LASLP2  042450R  002   LBUEXT  026310R  002
CSCCOM  010246R  002   CVLFLG= 020000          DVBTDA= 000060          LASLP3  042456R  002   LBUFBT  026236R  002
CSCERR  010474R  002   CVLMD1  012560R  002   DVCPRT= 000054          LASNIN  016077R  002   LBULP1  026046R  002
CSCFND= 000010         CVLMD2  012570R  002   DVCTEP= 000112          LASPAK  042530R  002   LBULP2  026162R  002
CSCFWD  010214R  002   CVLR3A  013234R  002   DVCVEC= 000070          LASZER  016076R  002   LBULP3  026170R  002
CSCN    010222R  002   CVMEND  001450R  002   DVGETB= 000076          LBA     043746R  002   LBULP4  026300R  002
CSCNBB  010234R  002   CVMST   001446R  002   DVIWSP= 000114          LBASE1  017334R  002   LBUMP1  036130R  002
CSCNCB  010234R  002   CVPER   010564R  002   DVNUM1  012742R  002   LBASE2  023624R  002   LBUMP2  036132R  002
CSCNCD  010242R  002   CVPNM1  010524R  002   DVPDTA= 000062          LBB     043752R  002   LBUMP3  036146R  002
CSCNEX  010472R  002   CVPNUM  010504R  002   DVPTEP= 000106          LBDDDN  041252R  002   LBUMP4  036150R  002
CSCNMS  010226R  002   CWA     001610R  002   DVPUTB= 000100          LBDEXT  041302R  002   LBUNBR  026326R  002
CSCNSP  010216R  002   CWRM    005660R  002   DVRID   012730R  002   LBDLIN  021274R  002   LBUSAS  026100R  002
CSCRLF= 100000         CWRMOK  005672R  002   DVRINT= 000074          LBDLP1  041226R  002   LBUSPC  026342R  002
CSCTCS= 000020         CYCDVL= 002000          DVSFWD= 000064          LBDLP2  041242R  002   LBUTRU  026330R  002
CSCTMS= 000040         CYCPRG= 040000          DVSVEC= 000066          LBDSFL  041266R  002   LBUWRD  026344R  002
CSDNRC  010770R  002   CZERO   005130R  002   DVTVEC= 000072          LBDSTP  041216R  002   LCARRY  036016R  002
CSFCKA  004620R  002   CZWMSG  010112R  002   DVUPRT= 000052          LBDTZF  041272R  002   LCBUF   016100R  002
CSLDEL  005020R  002   C1THKP  043566R  002   DVVTEP= 000110          LBIASD  041176R  002   LCKCR   015650R  002
CSPGUP  010574R  002   DECASC  041024R  002   ENTER   013640R  002   LBPAND  025504R  002   LCKOBL  027432R  002
CSPNFP  010600R  002   DECBIN  042116R  002   ERSTOP= 000004          LBPBS   025414R  002   LCK2WD  033562R  002
CSTDVI  013414R  002   DECBUF  042563R  002   FDVNAM  013546R  002   LBPERX  025772R  002   LCLBUF  015236R  002
CSTOP   002462R  002   DECTAD= 000042          FENTRY  027634R  002   LBPEXT  025762R  002   LCLEAR  022456R  002
CSUB    006106R  002   DELETE  000044R  002   FETCH   013734R  002   LBPLGP  025662R  002   LCMCTB  030652R  002
```

LCOBPT	027554R	002	LCPSWP	034130R	002	LDOINC	015410R	002	LFVODD	036166R	002	LLPCLS	014416R	002
LCOBUG	027552R	002	LCPTEC	034136R	002	LDONE	023246R	002	LFVOET	036160R	002	LLPCNH	014402R	002
LCOCNR	027576R	002	LCPTEN	033756R	002	LDRSTE	027360R	002	LGOEXT	016452R	002	LLPCOL	014050R	002
LCOEIT	027500R	002	LCPTGO	032702R	002	LDRTBS	027346R	002	LGOTO	022660R	002	LLPDDN	014172R	002
LCOGLT	027616R	002	LCPTG1	032706R	002	LDVDEM	045137R	002	LGTRTN	016060R	002	LLPDEL	014134R	002
LCOGPL	027606R	002	LCPTG2	032724R	002	LDVDER	044344R	002	LHINUM	041434R	002	LLPFEX	014756R	002
LCOLFD	027564R	002	LCPTG3	032744R	002	LDVDTB	044276R	002	LIFCLR	022504R	002	LLPFLI	014560R	002
LCOLP1	027444R	002	LCPTG4	033032R	002	LDVRIF	034124R	002	LIFEQ1	022526R	002	LLPGAL	014242R	002
LCOMB	027316R	002	LCPTG5	033042R	002	LENCFD	027740R	002	LIFEQ2	022625R	002	LLPGFC	013726R	002
LCOMO	042040R	002	LCPTG6	033056R	002	LENCMD	030014R	002	LIFGE1	022564R	002	LLPGLB	014400R	002
LCOM1	042042R	002	LCPTG7	033110R	002	LENCNT	027712R	002	LIFGT1	022540R	002	LLPIEN	014700R	002
LCOM2	042044R	002	LCPTG8	033160R	002	LEND	023166R	002	LIFLE1	022577R	002	LLPINL	014602R	002
LCOM3	042046R	002	LCPTG9	033230R	002	LENDEV	030026R	002	LIFLT1	022552R	002	LLPLNF	014270R	002
LCOM4	042050R	002	LCPTM1	034142R	002	LENDFD	027422R	002	LIFNE1	022612R	002	LLPLP1	014160R	002
LCOM5	042052R	002	LCPTOE	034000R	002	LENDFL	024310R	002	LIFNE2	022642R	002	LLPLP2	014224R	002
LCOM6	042054R	002	LCPTSF	033726R	002	LENEEX	030002R	002	LIFST	022464R	002	LLPLP3	014310R	002
LCOM7	042056R	002	LCPT10	033242R	002	LENFCF	027676R	002	LIGNOR	023250R	002	LLPLP4	014334R	002
LCOM8	042060R	002	LCPT11	033276R	002	LENLP1	027656R	002	LIITBS	027354R	002	LLPLP5	014372R	002
LCOM9	042062R	002	LCPT12	033306R	002	LENLP2	027720R	002	LIMIT	044640R	002	LLPLP6	014526R	002
LCOTG1	027456R	002	LCPT13	033324R	002	LENMSG	030006R	002	LINBUF	034262R	002	LLPLP7	014546R	002
LCOTG2	027626R	002	LCPT14	033342R	002	LENTER	016063R	002	LINCDG	015500R	002	LLPLP8	014572R	002
LCPABT	033700R	002	LCPT15	033360R	002	LENTRY	023166R	002	LINCDN	015502R	002	LLPMBH	014430R	002
LCPBTP	032574R	002	LCP67E	033150R	002	LERROR	023251R	002	LINCLN	015366R	002	LLPMDN	014536R	002
LCPBUG	032572R	002	LCRETN	042114R	002	LETGO	023166R	002	LINCR1	022411R	002	LLPMFC	014260R	002
LCPCNR	032460R	002	LCRLF	034430R	002	LEXIT	023166R	002	LINCR2	022413R	002	LLPMUP	014514R	002
LCPEFP	032342R	002	LCSTBL	030034R	002	LFETCH	015034R	002	LINDEX	027116R	002	LLPPAK	013766R	002
LCPEIT	032520R	002	LCTBST	030202R	002	LFGEXT	036324R	002	LINDIR	027076R	002	LLPPKB	014264R	002
LCPEMS	034146R	002	LCTCHR	026500R	002	LFGLP1	036300R	002	LINDTB	030620R	002	LLPRFL	015014R	002
LCPERL	034205R	002	LDASLP	041034R	002	LFGLP2	036304R	002	LINFND	027426R	002	LLPRPL	014436R	002
LCPERR	033670R	002	LDA1	041134R	002	LFILG	036274R	002	LININD	027054R	002	LLPRTF	037672R	002
LCPFCS	034144R	002	LDA2	041146R	002	LFILG1	023062R	002	LINNBR	016066R	002	LLPSNP	015036R	002
LCPFFC	032604R	002	LDA3	041160R	002	LFILG2	023127R	002	LIRTBS	027352R	002	LLPSOP	015040R	002
LCPFLN	033632R	002	LDECR1	022411R	002	LFILP	036326R	002	LIST	000050R	002	LLPTDN	014122R	002
LCPFTE	032476R	002	LDECR2	022413R	002	LFILP1	023077R	002	LIWS	027272R	002	LLPTEN	014640R	002
LCPGFC	032440R	002	LDECTB	041310R	002	LFILP2	023147R	002	LJSTFY	015726R	002	LLPTFF	014026R	002
LCPGNE	032432R	002	LDELAY	022411R	002	LFILR	036216R	002	LLIBCT	016302R	002	LLPTFS	014330R	002
LCPGNV	033366R	002	LDELET	023247R	002	LFILR1	023044R	002	LLIBPT	016304R	002	LLPWRN	014670R	002
LCPGTE	032666R	002	LDIADR	024662R	002	LFILR2	023106R	002	LLIEM1	016260R	002	LMCRTB	044644R	002
LCPILE	032264R	002	LDIGPA	024564R	002	LFILV	036152R	002	LLIERR	016230R	002	LMCRTE	045020R	002
LCPLP1	032306R	002	LDILCD	024610R	002	LFILV1	023077R	002	LLIERT	016240R	002	LMIDL	030624R	002
LCPMTX	034110R	002	LDILP1	024520R	002	LFILV2	023147R	002	LLIEW1	016112R	002	LMLMSG	015165R	002
LCPNAL	033534R	002	LDIMSG	024661R	002	LFINLN	027370R	002	LLIEW2	016114R	002	LMODFY	016062R	002
LCPNRF	034131R	002	LDIOBJ	024672R	002	LFLLP1	027374R	002	LLIEW3	016210R	002	LMOVE0	023166R	002
LCPNSW	033754R	002	LDIR	027136R	002	LFRCBC	036244R	002	LLIEW4	016212R	002	LMOVE1	022420R	002
LCPNT1	033012R	002	LDIREF	024650R	002	LFREE	030632R	002	LLIEXT	016250R	002	LMOVE2	022425R	002
LCPNT2	033220R	002	LDISP	027216R	002	LFREVN	036252R	002	LLINK	022660R	002	LMPNTR	021752R	002
LCPOBA	034140R	002	LDISVR	024502R	002	LFREXT	036272R	002	LLITBC	016144R	002	LMSTBE	021300R	002
LCPOCT	033520R	002	LDITBC	024540R	002	LFRODD	036230R	002	LLITMB	016172R	002	LNBRWD	026000R	002
LCPOEN	034132R	002	LDITEF	024550R	002	LFROET	036222R	002	LLJST1	016020R	002	LNEGAT	022411R	002
LCPOUT	034014R	002	LDLAY	036764R	002	LFTMSG	015046R	002	LLJUST	016002R	002	LNEMSG	015114R	002
LCPPIB	032472R	002	LDLLP1	036766R	002	LFVCBC	036174R	002	LLOAD	022402R	002	LNFMSG	015153R	002
LCPPRI	034034R	002	LDLLP2	036772R	002	LFVEVN	036202R	002	LLONUM	041436R	002	LNOTDE	040646R	002
LCPPTS	034134R	002	LDLMSG	015142R	002	LFVEXT	036214R	002	LLPADL	014344R	002	LNOTIT	026770R	002

LOADVR	000052R	002	LPKADP	021006R	002	LPKLP6	017236R	002	LPNEX	040012R	002	LROT2	023161R	002
LOBJLN	015042R	002	LPKAMP	017524R	002	LPKLP8	021014R	002	LPOADR	040232R	002	LRT1	031022R	002
LOCTSW	040400R	002	LPKBIT	020142R	002	LPKL25	016540R	002	LPOCBC	040250R	002	LRT10	031516R	002
LOCTTB	042326R	002	LPKBLP	020152R	002	LPKMPS	023262R	002	LPOCNT	040324R	002	LRT11	031530R	002
LOFF	016075R	002	LPKBT1	020516R	002	LPKNLN	016432R	002	LPOCR1	040316R	002	LRT12	031542R	002
LON	016074R	002	LPKBUF	023264R	002	LPKNXI	016674R	002	LPOCR2	040332R	002	LRT13	031554R	002
LPATA	036624R	002	LPKCBF	016334R	002	LPKOCT	017644R	002	LPOEVN	040256R	002	LRT14	031566R	002
LPATAP	042110R	002	LPKCBK	017064R	002	LPKPBT	020210R	002	LPOFXT	040276R	002	LRT15	031602R	002
LPATB	036652R	002	LPKCDE	017010R	002	LPKPKP	021272R	002	LPONOM	040322R	002	LRT16	031614R	002
LPATBP	042112R	002	LPKCEM	017402R	002	LPKPRT	020564R	002	LPOODD	040166R	002	LRT17	031626R	002
LPATCM	036662R	002	LPKCIO	016752R	002	LPKPSV	023260R	002	LPOOET	040160R	002	LRT18	031672R	002
LPATEN=	042114R	002	LPKCKT	017754R	002	LPKPTP	021270R	002	LPOPRT	040222R	002	LRT19	031704R	002
LPATEX	036666R	002	LPKCLN	016456R	002	LPKRFD	020470R	002	LPORTN	040326R	002	LRT2	031036R	002
LPATO	036342R	002	LPKCNB	017340R	002	LPKRFJ	020512R	002	LPOSDF	040136R	002	LRT20	031716R	002
LPATOP	042064R	002	LPKCRL	020352R	002	LPKSCC	016640R	002	LPRBSY	037664R	002	LRT21	031376R	002
LPAT1	036374R	002	LPKCRW	017120R	002	LPKSDL	016424R	002	LPRBUF	037674R	002	LRT22	031642R	002
LPAT1P	042066R	002	LPKCTE	016542R	002	LPKSDN	016410R	002	LPRCAL	037604R	002	LRT23	031656R	002
LPAT2	036426R	002	LPKCUE	017416R	002	LPKSEP	017260R	002	LPRCNT	037632R	002	LRT27	032106R	002
LPAT2P	042070R	002	LPKDEC	017704R	002	LPKSEV	017206R	002	LPRFWA	037636R	002	LRT28	032122R	002
LPAT3	036456R	002	LPKDER	017374R	002	LPKSEX	016564R	002	LPRNB	040402R	002	LRT29	032202R	002
LPAT3P	042072R	002	LPKEB1	016560R	002	LPKSIG	016416R	002	LPRNB1	022735R	002	LRT3	031410R	002
LPAT4	036530R	002	LPKEJ1	020132R	002	LPKSJ1	020120R	002	LPRNB2	023025R	002	LRT30	032262R	002
LPAT4P	042074R	002	LPKEJ2	020464R	002	LPKSMP	017312R	002	LPRND	040114R	002	LRT4	031422R	002
LPAT5	036542R	002	LPKEPT	023256R	002	LPKSPH	020056R	002	LPRND1	022720R	002	LRT5	031434R	002
LPAT5P	042076R	002	LPKERR	017430R	002	LPKSRS	020136R	002	LPRND2	023005R	002	LRT6	031446R	002
LPAT6	036554R	002	LPKEVN	021026R	002	LPKSR1	020230R	002	LPRNG	040022R	002	LRT7	031460R	002
LPAT6P	042100R	002	LPKEXT	017434R	002	LPKSR2	020250R	002	LPRNG1	022672R	002	LRT8	031472R	002
LPAT7	036566R	002	LPKFA1	021264R	002	LPKSR3	020262R	002	LPRNG2	022751R	002	LRT9	031504R	002
LPAT7P	042102R	002	LPKFA2	021260R	002	LPKSSP	017024R	002	LPRNI	022664R	002	LSAVLN	016042R	002
LPAT8	036600R	002	LPKFA3	021254R	002	LPKSTL	017444R	002	LPRNNR	022667R	002	LSBKUP	026552R	002
LPAT8P	042104R	002	LPKFA4	021250R	002	LPKSVP	017156R	002	LPRNO	040126R	002	LSCKGO	026574R	002
LPAT9	036612R	002	LPKFA5	021244R	002	LPKTAG	017742R	002	LPRNO1	022705R	002	LSCNCT	026626R	002
LPAT9P	042106R	002	LPKFA6	021240R	002	LPKTBE	021750R	002	LPRNO2	022767R	002	LSCNEX	026560R	002
LPAUSE	023166R	002	LPKFA7	021234R	002	LPKTBL	021302R	002	LPROGN	037756R	002	LSCNPT	026624R	002
LPBADR	040526R	002	LPKFIL	021052R	002	LPKTGA	020710R	002	LPROLN	015044R	002	LSCOUT	026556R	002
LPBASE	036336R	002	LPKFL1	021054R	002	LPKTGD	020020R	002	LPRRDE	037673R	002	LSET	022456R	002
LPBCBC	040544R	002	LPKFL2	021126R	002	LPKTG4	020532R	002	LRANDM	041324R	002	LSI11 =	000004	
LPBCFM	040536R	002	LPKFL3	021152R	002	LPKTG5	020554R	002	LRD72	015550R	002	LSKBL	026416R	002
LPBCR1	040576R	002	LPKFL4	021176R	002	LPKTG6	020560R	002	LREADY	015246R	002	LSKIP1	041070R	002
LPBEVN	040560R	002	LPKFL5	021206R	002	LPKTG7	020644R	002	LREAD1	023166R	002	LSKIP2	041110R	002
LPBLP1	040450R	002	LPKFT1	021062R	002	LPKTG8	020660R	002	LREAD2	022411R	002	LSMEND	030650R	002
LPBLP2	040470R	002	LPKIF	020302R	002	LPKTG9	020704R	002	LREAD3	023167R	002	LSNLP1	024404R	002
LPBODD	040436R	002	LPKIFD	016714R	002	LPKTNR	020774R	002	LREAD4	023176R	002	LSNTG1	024426R	002
LPBOET	040430R	002	LPKIFL	021276R	002	LPKTXD	021046R	002	LRELOC	043730R	002	LSNTG2	024440R	002
LPBRTN	040572R	002	LPKILU	016516R	002	LPKTXT	021002R	002	LRESET	023166R	002	LSPACE	016064R	002
LPBTG1	040502R	002	LPKITX	021012R	002	LPKUFC	017152R	002	LRETN	022662R	002	LSPFL	016010R	002
LPBTG2	040506R	002	LPKLIN	017454R	002	LPKVAR	017554R	002	LRJST2	015770R	002	LSPTST	015324R	002
LPGADR	040076R	002	LPKLPB	020044R	002	LPKWD1	020364R	002	LRLLP1	043744R	002	LSPTSV	026630R	002
LPGBYC	040100R	002	LPKLPG	020032R	002	LPKWD2	020424R	002	LRLLP2	044012R	002	LSTOBJ	024660R	002
LPGNCT	040104R	002	LPKLP1	016400R	002	LPLNNZ	024060R	002	LRLLP3	044044R	002	LSUB	036070R	002
LPGTG1	040062R	002	LPKLP3	016646R	002	LPLNTH	024311R	002	LROTR	036732R	002	LSUBCC	036064R	002
LPGTIL	040044R	002	LPKLP4	017046R	002	LPN	040014R	002	LROTRL	036752R	002	LSUBEX	036112R	002
LPHYSL	024312R	002	LPKLP5	017164R	002	LPNADR	040006R	002	LROT1	022411R	002	LSUBM	036020R	002

```
LSUBML  036050R  002   LTOO    040360R  002   LWRIT1  023166R  002   PSVREG= 000222           SREG    040776R  002
LSUB1   022435R  002   LTSTDE  040616R  002   LWRIT2  022411R  002   PTEM0  = 000056           STACK = 001200
LSUB2   022444R  002   LTTBIN  027012R  002   LWRIT3  023212R  002   PTEM1  = 000060           START   000064R  002
LSVBBC  025132R  002   LTTYCT  024460R  002   LWRIT4  023221R  002   PTEM10= 000102           STKBIG  035546R  002
LSVBCT  025306R  002   LUPAMP  024104R  002   LXINDT  030634R  002   PTEM11= 000104           STONER= 100000
LSVBCZ  025174R  002   LUPBIT  024154R  002   LXRTBS  027350R  002   PTEM12= 000106           STPTEX  036526R  002
LSVBNM  025142R  002   LUPCIX  016056R  002   LXTBL   030414R  002   PTEM13= 000110           STRPAT  036510R  002
LSVDWR  025246R  002   LUPCTA  024204R  002   L2CODF  034125R  002   PTEM14= 000112           SU4ROT  033154R  002
LSVEXT  025130R  002   LUPDLN  015560R  002   MASS70  034126R  002   PTEM15= 000114           SYNTAX  024320R  002
LSVFPS  025310R  002   LUPDL1  015632R  002   MAXPRG= 000020           PTEM2  = 000062           TAGLUP  026632R  002
LSVMDB  025060R  002   LUPDL2  015662R  002   MMMR3  = 172516           PTEM3  = 000064           TOEADR  034510R  002
LSVMLB  025006R  002   LUPEMB  024106R  002   MMVER  = 000001           PTEM4  = 000066           TSTVEC  040734R  002
LSVMOC  025162R  002   LUPEXT  024022R  002   MODIFY  013722R  002   PTEM5  = 000070           UFWCLR  013270R  002
LSVMRL  025270R  002   LUPIFD  023750R  002   MPGEND= 045212R  002   PTEM6  = 000072           UFWMEX  013326R  002
LSVPS2  024712R  002   LUPIMP  024276R  002   NOCOMP= 000001           PTEM7  = 000074           UFWSET  013252R  002
LSVRSW  024762R  002   LUPKPT  024314R  002   NPRFLG  034127R  002   PTEM8  = 000076           UFWTST  013306R  002
LSVRTL  025312R  002   LUPLIN  024174R  002   OCPRES= 000100           PTEM9  = 000100           ULCRLF  040304R  002
LSVTEN  025014R  002   LUPLP1  023420R  002   OCTBIN  042132R  002   PTEND  = 000242           ULIST   035640R  002
LSVUPL  024766R  002   LUPLP2  023504R  002   OPENL   000054R  002   PTLGTH= 000242           ULISTM  035622R  002
LSVWFT  025150R  002   LUPLP3  023650R  002   PACK    016306R  002   PTOCNT= 000030           ULSTCM  035650R  002
LSYMNF  027002R  002   LUPLP4  023756R  002   PASCIN= 000006           PTSIZE= 000240           UNPACK  023374R  002
LSYMTE  027356R  002   LUPNPI  023642R  002   PATCH   001652R  002   PUSRPC= 000236           UPCI    015224R  002
LSYNDT  027202R  002   LUPPTP  024316R  002   PATGEN  036670R  002   PUT     000056R  002     UPCRLF  040312R  002
LSYNIT  027152R  002   LUPSEP  023574R  002   PC     =%000007         PUTBYT  040772R  002     UPLI    016132R  002
LTAB    016072R  002   LUPSEV  023524R  002   PCURDV= 000035           PWRIOA= 000020           UPLIO   016104R  002
LTAGCT  027362R  002   LUPSEX  023670R  002   PDNUMS= 000036           RDIOSZ  012414R  002     UPRADR  035672R  002
LTBEND  023245R  002   LUPSIN  023774R  002   PDPNTR= 000034           REALNM  000062R  002     UPRASC  040026R  002
LTBERR  042302R  002   LUPSMP  023614R  002   PDST   = 000122           RJUSTI  015762R  002     UPRBYC  035674R  002
LTBFD   042224R  002   LUPTAG  024260R  002   PFBBOV= 000002           RREG    041010R  002     URSTOP= 000002
LTBIN0  042162R  002   LUPTGI  024206R  002   PFLGWD= 000000           RSTI    000070R  002     USMTPS= 000002
LTBIN1  042176R  002   LUPTMF  023630R  002   PFWADR= 000004           RST2    000124R  002     USRINT  000226R  002
LTBIN2  042200R  002   LUPVAR  024230R  002   PGCOMP  035566R  002   RTNINT  040764R  002     USRLP   037530R  002
LTBIN3  042236R  002   LVECDB  037344R  002   PGENEX  036730R  002   R0     =%000000           USRPEX  037656R  002
LTBIN4  042246R  002   LVECL1  037162R  002   PG6END= 042116R  002   R1     =%000001           USRPRT  037510R  002
LTBIN5  042264R  002   LVECL2  037226R  002   PLNGTH= 000026           R2     =%000002           USRPR1  037514R  002
LTBOUT  042310R  002   LVECMP  037274R  002   PMDLCD= 000032           R3     =%000003           VDFLSZ  042566R  002
LTBSAV  042146R  002   LVECT   023233R  002   PNAME  = 000010           R4     =%000004           VECEX   040756R  002
LTGADR  027364R  002   LVEDEP  037166R  002   PNBR   = 000116           R5     =%000005           WRIOSZ  012424R  002
LTGLN1  026756R  002   LVEHDN  037416R  002   POBJST= 000024           R6     =%000006           WT4IOT= 000010
LTGLN2  026746R  002   LVEHDR  037372R  002   POPSW  = 000002           R7     =%000007           XCALC   027126R  002
LTLEP2  026656R  002   LVEMSG  037456R  002   PRDIOA= 000016           SAVE    024702R  002     XXXX   = 000000
LTLEP3  026670R  002   LVEREX  037256R  002   PRONER= 020000           SCAN    026406R  002     ZERO    000060R  002
LTLEXT  027334R  002   LVERFY  037004R  002   PSRC   = 000120           SETDED= 000040           .      = 045212R  002

. ABS.  000000  000
        000000  001
MPG     045212  002

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DTMGAC/NL:TOC/DOC=DTMGAC.P11
 RUN-TIME: 35 71 4 SECONDS
 RUN-TIME RATIO: 141/111=1.2
 CORE USED:  8K  (15 PAGES)
```

DOCUMENT PAGES: 246