RP02/03

DEVICE ROUTINE (MPG)

MD-11-DTR3A-A

EP-DTR3A-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A

```
                        .REM    %


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

                        PRODUCT CODE:   MAINDEC-11-DTR3A-A
                        PRODUCT NAME:   RP02/RP03 DEVICE ROUTINE FOR MPG
                        DATE:           JULY 1976
                        MAINTAINER:     SYSTEMS RELIABILITY
                        AUTHOR:         C. E. HARPER
```



```
                        %
```

```
                                        .SBTTL  REVISION HISTORY

                                ;       JUL 76  DTR3A-A INITIAL RELEASE
```

```
                                        .SBTTL  STANDARD DEVICE ROUTINE TABLE
                                        .TITLE  MAINDEC-11-DTR3A-A  RP02/RP03 DEVICE ROUTINE FOR MPG
                                        ;REVISION 'A'
                                        ;FILENAME OF "TR3AAD.MPG" ON MPG/XXDP MEDIA
                                        ;MACY11:  DTR3A?,DTR3A?/CRF:SYM/DOC=DTR3A?.P11
                                        ;LNKX11:  DTR3A?.MPG/B:0-DTR3A?/E
                                        ;PAPER TAPE:  PUNCH DTR3A?.MPG/FILE:ELEV

            000000'                     .CSECT  RP11
                                        .DSABL  GBL

                                        ;THE FOLLOWING TABLE IS IN THE STANDARDIZED FORMAT REQUIRED
                                        ;TO INTERFACE WITH MPG.

000000' 007402              LOCZ:       .WORD   DVREND-.            ;DEVICE ROUT SIZE IN BYTES
000002' 000000              DFLGWD:     .WORD   0                   ;DEVICE ROUT FLAGWORD
                                        WAITMD= 100000              ;  WAIT MODE - 0 = WAIT
                                        WAITTN= 4                   ;  WAITNG FOR ATTN INT
                                        DOTERM= 2                   ;  PROCESS I/O TERMINATION
                                        IOERR=  1                   ;  ERROR ON CURRENT I/O
000004' 000000              CYL:        .WORD   0                   ;CYLINDER # (0 THRU 202./405.)
000006' 000000              HEAD:       .WORD   0                   ;HEAD # (0 THRU 19.)
000010' 000000              SECT:       .WORD   0                   ;SECTOR # (0 THRU 9.)
000012' 000003              RTRY:       .WORD   3                   ;# OF RETRY ATTEMPTS
000014' 000000                          .WORD   0                   ;INTERFACE WORD # 5 (NOT USED)
000016' 000000                          .WORD   0                   ;INTERFACE WORD # 6 (NOT USED)
000020' 000001              SIZE:       .WORD   1                   ;# OF BYTES TRANSFERRED / UNIMAP FLG
000022' 000000              ERRI:       .WORD   0                   ;ERROR ON LAST I/O INDICATOR
000024' 176710              DREGAD:     .WORD   176710              ;FIRST DEVICE REGISTER ADR
000026' 000254              IVCTAD:     .WORD   254                 ;INTERRUPT VECTOR ADR
000030' 000240              PSWD:       .WORD   240                 ;INT PROC STATUS WORD (BR 5)
000032' 000000                          .WORD   0                   ;NOT USED
000034' 001330                          .WORD   HSKEEP-.            ;HOUSEKEEPING ROUT REL ADR
000036' 001400                          .WORD   REPORT-.            ;REPORT ROUT REL ADR
000040' 002014                          .WORD   KILL-.              ;KILL ROUT REL ADR
000042' 001254                          .WORD   DATAER-.            ;DATA ERROR COUNTER REL ADR
000044' 001730                          .WORD   TOUTER-.            ;TIME OUT ERROR ROUT REL ADR
000046' 000000              CIOBSY:     .WORD   0                   ;I/O BUSY BRANCH ADR
000050' 000000              CUPGER:     .WORD   0                   ;DEVICE ERROR BRANCH ADR
000052' 000000              ULIST:      .WORD   0                   ;USER MODE PRINT ROUTINE BRANCH ADR
000054' 000000              CLIST:      .WORD   0                   ;CMND MODE PRINT ROUTINE BRANCH ADR
000056' 000000              BINASC:     .WORD   0                   ;CONVERT BINARY TO ASCII ROUT BR ADR
000060' 000000              BTASLZ:     .WORD   0                   ;CONVERT BINARY TO DECIMAL ASCII BR ADR
000062' 000000              DECASC:     .WORD   0                   ;CONVERT PACKED DECIMAL TO ASCII BR ADR
000064' 000000              CSYSFW:     .WORD   0                   ;MPG SYSTEM FLAGWORD ADR
000066' 000000              SETVEC:     .WORD   0                   ;SET INT VECT ROUT BR ADR
000070' 000000              CLRVEC:     .WORD   0                   ;CLEAR INT VECTOR ROUT BR ADR
000072' 000000              TSTVEC:     .WORD   0                   ;TEST INT VECTOR ROUT BR ADR
000074' 000000              RTNINT:     .WORD   0                   ;RETURN FROM INT ROUT BR ADR
```

```
                                GETBYT: .WORD   0                       ;GET DATA BYTE ROUT BR ADR
                                PUTBYT: .WORD   0                       ;PUT DATA BYTE ROUT BR ADR
                                        .WORD   DVREGS-.                ;ADR OF DEVICE REGISTER NAMES
                                        .WORD   DVCMDS-.                ;ADR OF DEVICE FUNCTIONS
                                        .WORD   DVPKTE-.                ;ADR OF PACK TBL EXTENSION
                                        .WORD   DVMVTE-.                ;ADR OF MODEL VECTOR TBL EXTEN.
                                        .WORD   DVCPTE-.                ;ADR OF COMPILER TBL EXTEN.
                                        .WORD   DVIWST-.                ;ADR OF DEV INTERFACE WD SYM TBL
```

```
                                        .SBTTL  COMPILER TABLES & CONSTANT AREAS

000116' 050122 051504   DVREGS: .ASCII  /RPDS/                  ;VALID DEVICE REGISTER NAMES &
000122' 000000                  .WORD   0                       ;THEIR POSITIONS RELATIVE TO
000124' 050122 051105           .ASCII  /RPER/                  ;THE DEVICE REGISTERS BASE ADDRESS.
000130' 000002                  .WORD   2
000132' 050122 051503           .ASCII  /RPCS/
000136' 000004                  .WORD   4
000140' 050122 041527           .ASCII  /RPWC/
000144' 000006                  .WORD   6
000146' 050122 040502           .ASCII  /RPBA/
000152' 000010                  .WORD   10
000154' 050122 040503           .ASCII  /RPCA/
000160' 000012                  .WORD   12
000162' 050122 040504           .ASCII  /RPDA/
000166' 000014                  .WORD   14
000170' 050122 030515           .ASCII  /RPM1/
000174' 000016                  .WORD   16
000176' 050122 031115           .ASCII  /RPM2/
000202' 000020                  .WORD   20
000204' 050122 031515           .ASCII  /RPM3/
000210' 000022                  .WORD   22
000212' 052523 040503           .ASCII  /SUCA/
000216' 000024                  .WORD   24
000220' 044523 047514           .ASCII  /SILO/
000224' 000026                  .WORD   26
        000226'         DVREGE= .

000226'     120     201 DVCMDS: .BYTE   120,201                 ;VALID DEVICE FUNCTIONS
000230' 002232                  .WORD   READ-.                  ;  FLAG BYTE:
000232'     130     201         .BYTE   130,201                 ;    BIT 7 = NPR DEV
000234' 002256                  .WORD   WRITE-.                 ;    BIT 3 = MASSBUS DEV
000236'     376     000         .BYTE   376,0                   ;    BIT 0 = 2 WORDS FOR ADR
000240' 002072                  .WORD   NOWAIT-.                ;            (18 BIT ADRS)
000242'     375     000         .BYTE   375,0
000244' 002046                  .WORD   WAIT-.
000246'     374     000         .BYTE   374,0
000250' 001166                  .WORD   REPORT-.
000252'     373     000         .BYTE   373,0
000254' 001162                  .WORD   REPORT-.
000256'     372     000         .BYTE   372,0
000260' 001632                  .WORD   STEPUP-.
000262'     371     000         .BYTE   371,0
000264' 001760                  .WORD   STEPDN-.
000266'     370     201         .BYTE   370,201
000270' 002252                  .WORD   RDNOSK-.
000272'     367     201         .BYTE   367,201
000274' 002260                  .WORD   WRNOSK-.
000276'     366     201         .BYTE   366,201
000300' 002266                  .WORD   WRCK-.
000302'     365     000         .BYTE   365,0
000304' 002312                  .WORD   SEEK-.
000306'     364     000         .BYTE   364,0
000310' 002330                  .WORD   HOMESK-.
```

```
000312' 000363  000                     .BYTE   363,0
000314' 002324                          .WORD   RECAL-.
000316' 000362  000                     .BYTE   362,0
000320' 002062                          .WORD   IDLE-.
000322' 000361  000                     .BYTE   361,0
000324' 002056                          .WORD   CRESET-.
000326' 000360  000                     .BYTE   360,0
000330' 002012                          .WORD   HORON-.
000332' 000357  000                     .BYTE   357,0
000334' 002016                          .WORD   HOROFF-.
000336' 000356  000                     .BYTE   356,0
000340' 002022                          .WORD   MODE11-.
000342' 000355  000                     .BYTE   355,0
000344' 002026                          .WORD   MODE10-.
000346' 177777                          .WORD   177777          ;TABLE TERMINATOR

000350' 047516 040527 052111 DVPKTE:    .ASCII  /NOWAIT/        ;PACK TABLE EXTENSION
000356' 000376  000                     .BYTE   376,0
000360' 020040 040527 052111            .ASCII  /  WAIT/
000366' 000375  000                     .BYTE   375,0
000370' 052123 052101 051525            .ASCII  /STATUS/
000376' 000374  000                     .BYTE   374,0
000400' 047503 047125 051524            .ASCII  /COUNTS/
000406' 000373  000                     .BYTE   373,0
000410' 052123 050105 050125            .ASCII  /STEPUP/
000416' 000372  000                     .BYTE   372,0
000420' 052123 050105 047104            .ASCII  /STEPDN/
000426' 000371  000                     .BYTE   371,0
000430' 042122 047516 045523            .ASCII  /RDNOSK/
000436' 000370  000                     .BYTE   370,0
000440' 051127 047516 045523            .ASCII  /WRNOSK/
000446' 000367  000                     .BYTE   367,0
000450' 020040 051127 045503            .ASCII  /  WRCK/
000456' 000366  000                     .BYTE   366,0
000460' 020040 042523 045505            .ASCII  /  SEEK/
000466' 000365  000                     .BYTE   365,0
000470' 047510 042515 045523            .ASCII  /HOMESK/
000476' 000364  000                     .BYTE   364,0
000500' 051040 041505 046101            .ASCII  / RECAL/
000506' 000363  000                     .BYTE   363,0
000510' 020040 042111 042514            .ASCII  /  IDLE/
000516' 000362  000                     .BYTE   362,0
000520' 051103 051505 052105            .ASCII  /CRESET/
000526' 000361  000                     .BYTE   361,0
000530' 044040 051104 047117            .ASCII  / HORON/
000536' 000360  000                     .BYTE   360,0
000540' 042110 047522 043106            .ASCII  /HOROFF/
000546' 000357  000                     .BYTE   357,0
000550' 047515 042504 030461            .ASCII  /MODE11/
000556' 000356  000                     .BYTE   356,0
000560' 047515 042504 030061            .ASCII  /MODE10/
000566' 000355  000                     .BYTE   355,0

000570' 000376 001140 DVMVTE:           .WORD   376,MSFMT1-LOCZ         ;MODEL VECTOR TABLE EXTEN.
000574' 000375 001140                   .WORD   375,MSFMT1-LOCZ
000600' 000374 001140                   .WORD   374,MSFMT1-LOCZ
```

```
000604' 000373  001140                  .WORD   373,MSFMT1-LOCZ
000610' 000372  001164                  .WORD   372,MSFMT5-LOCZ
000614' 000371  001164                  .WORD   371,MSFMT5-LOCZ
000620' 000370  001146                  .WORD   370,MSFMT3-LOCZ
000624' 000367  001155                  .WORD   367,MSFMT4-LOCZ
000630' 000366  001141                  .WORD   366,MSFMT2-LOCZ
000634' 000365  001140                  .WORD   365,MSFMT1-LOCZ
000640' 000364  001140                  .WORD   364,MSFMT1-LOCZ
000644' 000363  001140                  .WORD   363,MSFMT1-LOCZ
000650' 000362  001140                  .WORD   362,MSFMT1-LOCZ
000654' 000361  001140                  .WORD   361,MSFMT1-LOCZ
000660' 000360  001140                  .WORD   360,MSFMT1-LOCZ
000664' 000357  001140                  .WORD   357,MSFMT1-LOCZ
000670' 000356  001140                  .WORD   356,MSFMT1-LOCZ
000674' 000355  001140                  .WORD   355,MSFMT1-LOCZ

                                ;
                                ;       COMPILER TABLE EXTENSION
                                ;
000700'     003     376         DVCPTE: .BYTE   3,376                   ;NO WAIT
000702' 004537  000012                  .WORD   4537,10.
000706'     003     375                 .BYTE   3,375                   ;WAIT
000710' 004537  000012                  .WORD   4537,10.
000714'     004     374                 .BYTE   4,374                   ;STATUS
000716' 004537  000012  001002          .WORD   4537,10.,1002
000724'     004     373                 .BYTE   4,373                   ;COUNTS
000726' 004537  000012  001001          .WORD   4537,10.,1001
000734'     004     372                 .BYTE   4,372                   ;STEP UP
000736' 004537  000012  000000          .WORD   4537,10.,0
000744'     004     371                 .BYTE   4,371                   ;STEP DOWN
000746' 004537  000012  000000          .WORD   4537,10.,0
000754'     006     370                 .BYTE   6,370                   ;READ NO SEEK
000756' 004537  000012  000000          .WORD   4537,10.,0,2,2
000764' 000002  000002
000770'     006     367                 .BYTE   6,367                   ;WRITE NO SEEK
000772' 004537  000012  000000          .WORD   4537,10.,0,2,2
001000' 000002  000002
001004'     006     366                 .BYTE   6,366                   ;WRITE CHECK
001006' 004537  000012  000000          .WORD   4537,10.,0,2,2
001014' 000002  000002
001020'     003     365                 .BYTE   3,365                   ;SEEK
001022' 004537  000012                  .WORD   4537,10.
001026'     003     364                 .BYTE   3,364                   ;HOME SEEK
001030' 004537  000012                  .WORD   4537,10.
001034'     003     363                 .BYTE   3,363                   ;RECAL
001036' 004537  000012                  .WORD   4537,10.
001042'     003     362                 .BYTE   3,362                   ;IDLE
001044' 004537  000012                  .WORD   4537,10.
001050'     003     361                 .BYTE   3,361                   ;CRESET
001052' 004537  000012                  .WORD   4537,10.
001056'     003     360                 .BYTE   3,360                   ;HEADER ON
001060' 004537  000012                  .WORD   4537,10.
001064'     003     357                 .BYTE   3,357                   ;HEADER OFF
001066' 004537  000012                  .WORD   4537,10.
001072'     003     356                 .BYTE   3,356                   ;MODE 11
001074' 004537  000012                  .WORD   4537,10.
```

```
001100' 003 355                 .BYTE   3,355                   ;MODE 10
001102' 004537 000012           .WORD   4537,10.

                        ;
                        ;       DEVICE INTERFACE WORD SYMBOL TABLE
                        ;
001106' 054503 020114   DVIWST: .ASCII  /CYL /
001112' 000004                  .WORD   DEVIW1
001114' 042510 042101           .ASCII  /HEAD/
001120' 000006                  .WORD   DEVIW2
001122' 042523 052103           .ASCII  /SECT/
001126' 000010                  .WORD   DEVIW3
001130' 052122 054522           .ASCII  /RTRY/
001134' 000012                  .WORD   DEVIW4
001136' 177777                  .WORD   177777                  ;END OF TABLE

                        ;
                        ;       MODEL STATEMENT TABLE EXTENSION
                        ;
001140'     000         MSFMT1: .BYTE   0
001141'     377 052101 000377   MSFMT2: .ASCIZ  <377>/AT/<377>
001146' 044777 052116 177517    MSFMT3: .ASCIZ  <377>/INTO/<377>
001154'     000
001155'     377 051106 046517   MSFMT4: .ASCIZ  <377>/FROM/<377>
001162' 000377
001164'     377     000         MSFMT5: .BYTE   377,0
                                .EVEN


                                ;DEVICE ROUTINE CONSTANTS & EQUATES


        001166'                 HSKPST= .
        001166'                 ISTAT=  .                       ;STORAGE FOR DEV REG'S AT INT
001166' 000000 000000 000000            .WORD   0,0,0,0,0,0
001174' 000000 000000 000000
001202' 000000 000000 000000            .WORD   0,0,0,0,0
001210' 000000 000000

001214' 000013          CSTAT:  .BLKW   11.                     ;DEV REG CURRENT VALUES STORAGE

001242' 000000          OBJADR: .WORD   0                       ;CURR ADR IN USER PROG
001244' 000000          CNTADR: .WORD   0                       ;ADR OF BYTE COUNT TOTALS
001246' 000000          CURFLG: .WORD   0                       ;FLAG WORD OF CURR CMND
001250' 000000          CURCMD: .WORD   0                       ;CURR CMND CODE
001252' 000000          CURADR: .WORD   0                       ;CURR BUS ADDRESS
001254' 000000                  .WORD   0
001256' 000000          CURCNT: .WORD   0                       ;NEG WORD CNT FOR CURR CMND
001260' 000000          FINCNT: .WORD   0                       ;FINAL WORD CNT (RPWC)
001262' 000000          CURRTY: .WORD   0                       ;CURR RETRY COUNT
001264' 000000          RTRYIP: .WORD   0                       ;RETRY IN PROGRESS FLAG

001266'                 COUNTS:
001266' 000000          BYRD:   .WORD   0                       ;BYTES READ COUNT
001270' 000000                  .WORD   0
001272' 000000          BYWR:   .WORD   0                       ;BYTES WRITTEN COUNT
```

```
001274' 000000                  .WORD   0
001276' 000000          BYCK:   .WORD   0               ;BYTES CHECKED COUNT
001300' 000000                  .WORD   0
001302' 000000          RDCNT:  .WORD   0               ;READ CMND COUNT
001304' 000000          WRCNT:  .WORD   0               ;WRITE CMND COUNT
001306' 000000          CKCNT:  .WORD   0               ;CHECK CMND COUNT
001310' 000000          SKCNT:  .WORD   0               ;SEEK CMND COUNT
001312' 000000          MISCNT: .WORD   0               ;MISC. CMND COUNT
001314' 000000          ERRCNT: .WORD   0               ;DEVICE ERRORS COUNT
001316' 000000          DATAER: .WORD   0               ;DATA ERRORS COUNT
001320' 000000          TIECNT: .WORD   0               ;TIMING ERRORS (DLT)
001322' 000000          CSECNT: .WORD   0               ;CHECKSUM ERRORS
001324' 000000          WPECNT: .WORD   0               ;WORD PARITY ERRORS
001326' 000000          LPECNT: .WORD   0               ;LONGITUDINAL PARITY ERRORS
001330' 000000          WCECNT: .WORD   0               ;WRITE CHECK ERRORS
001332' 000000          RETRYS: .WORD   0               ;# OF RETRIES ON I/O CMNDS
001334' 000000          INTCNT: .WORD   0               ;INTERRUPTS COUNT

        001336'         HSKPEN= .

001336' 000000          RPCSV:  .WORD   0               ;BASE VALUE FOR RPCS REG

        010000                  MODE=   10000           ;  MODE SELECT BIT - 1 = PDP-10/15
        004000                  HEADER= 4000            ;  HEADER BIT - 1 = HEADER OPERATION

                                ;RPCS                   DEVICE BIT EQUATES

        100000                  ERR=    100000
        040000                  HE=     40000

        000013          REGNUM= 11.                     ;# OF DEVICE REGISTERS
        000024          CNTNUM= HSKPEN-COUNTS/2         ;# OF STATISTICAL COUNT WORDS

        000001          MMVER=  1                       ;SYS FLGWD BIT DEFINITIONS

        000000          XXXX=   0                       ;VALUE TO BE TAILORED BY DEV ROUT

        177774          RPDS=   -4                      ;EQUATES FOR DEVICE REGISTER NAMES
        177776          RPER=   -2                      ;RELATIVE TO R4'S CONTENTS
        000000          RPCS=    0
        000002          RPWC=   +2
        000004          RPBA=   +4
        000006          RPCA=   +6
        000010          RPDA=   +10
        000012          RPM1=   +12
        000014          RPM2=   +14
        000016          RPM3=   +16
        000020          SUCA=   +20
        000022          SILO=   +22

001340'                 PATCH:  .REPT   10.             ;PATCH AREA
                                .WORD   0
                                .ENDR
```

```
                                        .SBTTL  RP02/RP03 SUPPORT ROUTINES ENTERED FROM MPG


                                        ;DEVICE ROUTINE HOUSEKEEPING

                                                ;JSR    R5,HSKEEP       S/R CALL
                                                ;.WORD  0 OR 1          0 = DO HSKP PER OPSW
                                                ;                       1 = UNCOND. DO HSKP
                                                ;
                                                ;R2 = PROG'S OPSW
                                                ;
                                                ;DESTROYS R0,R1

001364' 012767  000003  176420  HSKEEP: MOV     #3,RTRY                 ;INIT # OF RETRY ATTEMPTS
001372' 005067  177740                  CLR     RPCSV                   ;INITIALIZE RPCS VALUE
001376' 010700                          MOV     PC,R0                   ;SET UP FIRST WD ADR
001400' 062700  177566                  ADD     #HSKPST-.,R0
001404' 012701  000064                  MOV     #HSKPEN-HSKPST/2,R1     ;SET UP # OF WORDS
001410' 005725                          TST     (R5)+                   ;UNCONDITIONALLY DO HSKP?
001412' 001005                          BNE     10$                     ;N,Y-10$
001414' 032702  000004                  BIT     #HSKPEP,R2              ;OPSW SPECIFY DON'T HSKP COUNTS?
001420' 001402                          BEQ     10$                     ;Y,N-10$
001422' 162701  000024                  SUB     #CNTNUM,R1              ;REMOVE THEM FROM LOOP COUNT
001426' 005020                  10$:     CLR     (R0)+                  ;HSKP ALL NECESSARY AREAS
001430' 005301                          DEC     R1
001432' 001375                          BNE     10$
001434' 000205                          RTS     R5                      ;EXIT IN-LINE


                                        ;RP02/RP03 REPORT ROUTINE

                                                ;JSR    R5,REPORT       S/R CALL
                                                ;.WORD  FLGWD           FLAGWORD
                                                ;                         BIT 15 = CMND MODE CALL
                                                ;                         BIT  9 = PROG STMNT CALL
                                                ;                         BIT  1 = DO STATUS REPORT
                                                ;                         BIT  0 = DO COUNTS REPORT

001436' 004067  003704  REPORT: JSR     R0,SAVREG               ;SAVE REG'S R0 - R5
001442' 004767  003732          JSR     PC,SUPTAD               ;SET UP PROG TBL ADR IN R3
001446' 011504                  MOV     (R5),R4                 ;GET FLAGWORD
001450' 032704  000002          BIT     #2,R4                   ;GOING TO DO STATUS DISPLAY?
001454' 001403                  BEQ     5$                      ;Y,N-5$
001456' 004567  003742          JSR     R5,STSTAT               ;GO STORE STATUS REG'S
001462' 177532                  .WORD   CSTAT-.
001464' 032704  177776  5$:      BIT     #177776,R4              ;DISPLAYING CNTS AT END OF
001470' 001012                  BNE     15$                     ;PROG PASS? (Y,N-15$)
001472' 010700                  MOV     PC,R0                   ;SET UP ADR OF CNTS
001474' 062700  177572          ADD     #COUNTS-.,R0
001500' 012701  000024          MOV     #CNTNUM,R1              ;GET # OF CNT WORDS
001504' 005720          10$:    TST     (R0)+                   ;THIS CNT WORD = 0?
001506' 001003                  BNE     15$                     ;Y,N-15$
001510' 005301                  DEC     R1                      ;DECR WORD CNT
001512' 001374                  BNE     10$                     ;CK'ED ALL WORDS? (Y,N-10$)
001514' 000477                  BR      DVREX                   ;GO TO EXIT -- ALL CNTS ARE 0'S
001516' 004767  003726  15$:    JSR     PC,DEVID                ;DISPLAY DEVICE I.D.
001522' 032704  000002          BIT     #2,R4                   ;DOING STATUS DISPLAY?
```

```
                                        ;
                                        ;
001526' 001432                  BEQ     DISCNT          ;Y,N-DISCNT
001530' 010700                  MOV     PC,R0           ;SET UP ADR OF REG'S AT
001532' 062700  177434          ADD     #ISTAT-.,R0     ;LAST INT
001536' 012701  000013          MOV     #REGNUM,R1      ;SET UP # OF REG'S
001542' 005720          20$:    TST     (R0)+           ;ALL REG'S = 0?
001544' 001003                  BNE     30$             ;N,Y-40$
001546' 005301                  DEC     R1
001550' 001374                  BNE     20$
001552' 000407                  BR      40$
001554' 004567  004240  30$:    JSR     R5,PRINT        ;ISSUE 'AT LAST INT' MSG
001560' 004361                  .WORD   ATIMSG-.
001562' 000014                  .WORD   12.
001564' 004567  003774          JSR     R5,DISPST       ;GO DISPLAY STATUS AT LAST INT
001570' 177376                  .WORD   ISTAT-.
001572' 004567  004222  40$:    JSR     R5,PRINT        ;ISSUE 'CURRENTLY' MSG
001576' 004357                  .WORD   CURMSG-.
001600' 000012                  .WORD   10.
001602' 004567  003756          JSR     R5,DISPST       ;GO DISPLAY CURRENT STATUS
001606' 177406                  .WORD   CSTAT-.
001610' 004767  004134          JSR     PC,PRTIWD       ;GO DISPLAY INFO WORDS
001614' 032704  000001  DISCNT: BIT     #1,R4           ;DISPLAY COUNTS?
001620' 001431                  BEQ     RPTEND          ;Y,N-RPTEND
001622' 012700  000024          MOV     #CNTNUM,R0      ;SET UP # OF WORDS
001626' 010701                  MOV     PC,R1           ;SET UP ADR OF CNTS
001630' 062701  177436          ADD     #COUNTS-.,R1
001634' 010702                  MOV     PC,R2           ;SET UP TBL ADR
001636' 062702  000066          ADD     #REPTBL-.,R2
001642' 012267  000012  RPTLP:  MOV     (R2)+,RPTBAS    ;MOV MSG ADR TO S/R LINKAGE
001646' 004067  003474          JSR     R0,SAVREG       ;SAVE ALL REG'S
001652' 011100                  MOV     (R1),R0         ;GET CURRENT COUNT
001654' 004577  176176          JSR     R5,@BINASC      ;CONVERT IT TO ASCII
001660' 000000  RPTBAS: .WORD   XXXX
001662' 004067  003474          JSR     R0,RESREG       ;RESTORE REG'S
001666' 005721                  TST     (R1)+           ;POINT AT NXT CNT
001670' 005300                  DEC     R0              ;DONE ALL WORDS?
001672' 001363                  BNE     RPTLP           ;Y,N-RPTLP
001674' 004567  004120          JSR     R5,PRINT        ;GO ISSUE COUNTS MSG
001700' 004406                  .WORD   CNTSMG-.
001702' 000440                  .WORD   CNTSEN-CNTSMG
001704' 004567  004110  RPTEND: JSR     R5,PRINT        ;ISSUE "END OF REPORT" MSG
001710' 004257                  .WORD   RENDMG-.
001712' 177763                  .WORD   -13.
001714' 004067  003442  DVREX:  JSR     R0,RESREG       ;RESTORE REGISTERS
001720' 005725                  TST     (R5)+           ;SET UP RETURN POINT
001722' 000205                  RTS     R5              ;EXIT IN-LINE
                                        ;
                                        ;
001724' 004442          REPTBL: .WORD   BCMRD-RPTBAS
001726' 004450                  .WORD   BCMRD+6-RPTBAS
001730' 004464                  .WORD   BCMWR-RPTBAS
001732' 004472                  .WORD   BCMWR+6-RPTBAS
001734' 004510                  .WORD   BCMCK-RPTBAS
001736' 004516                  .WORD   BCMCK+6-RPTBAS
001740' 004543                  .WORD   CMDCRD-RPTBAS
001742' 004556                  .WORD   CMDCWR-RPTBAS
001744' 004571                  .WORD   CMDCCK-RPTBAS
```

```
001746' 004607                  .WORD   CMDCSK-RPTBAS
001750' 004624                  .WORD   CMDCMS-RPTBAS
001752' 004652                  .WORD   CNTERR-RPTBAS
001754' 004667                  .WORD   CNTDER-RPTBAS
001756' 004717                  .WORD   CNTTIE-RPTBAS
001760' 004734                  .WORD   CNTCSE-RPTBAS
001762' 004750                  .WORD   CNTNXE-RPTBAS
001764' 004767                  .WORD   CNTLPE-RPTBAS
001766' 005003                  .WORD   CNTWCE-RPTBAS
001770' 005032                  .WORD   CNTRTY-RPTBAS
001772' 005060                  .WORD   CNTINT-RPTBAS


                        ;TIMEOUT ERROR ROUTINE

                                ;JSR     R5,TOUTER       S/R CALL

001774' 004067  003346  TOUTER: JSR     R0,SAVREG       ;SAVE ALL REGISTERS
002000' 004767  003374          JSR     PC,SUPTAD       ;SET UP RPCS1 & PROG TBL ADR'S
002004' 004567  003414          JSR     R5,STSTAT       ;STORE CURRENT STATUS
002010' 177204                  .WORD   CSTAT-.
002012' 004567  002332          JSR     R5,TVECT        ;CK IF I HAVE VECTOR CONTROL
002016' 000404                  BR      10$             ;BR IF I DON'T
002020' 042714  020100          BIC     #20100,(R4)     ;RESET BOTH INT ENABLES
002024' 004767  002274          JSR     PC,RINTV        ;RESET THE INTERRUPT VECTOR
002030' 042713  000010  10$:    BIC     #WT4IOT,(R3)    ;RESET WAITING FOR I/O FLG
002034' 004567  002346          JSR     R5,ERRCS1       ;ISSUE I/O TIMEOUT ERROR MSG
002040' 002441                  .WORD   IOTO-ERMBAS
002042' 004067  003314          JSR     R0,RESREG       ;RESTORE REGISTERS
002046' 012605                  MOV     (SP)+,R5        ;REMOVE RETURN ADR
002050' 000177  175774          JMP     @CUPGER         ;GO TO ERROR EXIT


                        ;KILL USER PROGRAM ROUTINE

                                ;JSR     R5,KILL         S/R CALL
                                ;
                                ;R3 = PROG TBL ADR
                                ;
                                ;DESTROYS R0,R1

002054' 016701  175744  KILL:   MOV     DREGAD,R1       ;GET DEV REG ADR
002060' 062701  000004          ADD     #4,R1           ;POINT AT RPCS REG
002064' 004567  002260          JSR     R5,TVECT        ;DO I HAVE VECTOR CONTROL?
002070' 000407                  BR      KILLEX          ;BR IF I DON'T
002072' 032711  020100          BIT     #20100,(R1)     ;ARE INT ENABLES SET?
002076' 001402                  BEQ     10$             ;Y,N-10$
002100' 042711  020100          BIC     #20100,(R1)     ;RESET INT ENABLES
002104' 004767  002214  10$:    JSR     PC,RINTV        ;RESET INT VECTOR INFO
002110' 000205          KILLEX: RTS     R5              ;EXIT IN-LINE
```

```
                                        .SBTTL  RP02/RP03 NON-I/O & NON-INTERRUPT FUNCTION ROUTINES
                                        ;"STEPUP" FUNCTION ROUTINE
                                                ;JSR    R5,STEPUP       FUNCTION CALL
                                                ;.WORD  NBR             INCREMENT FACTOR
002112' 004767  000060          STEPUP: JSR     PC,STPCOM               ;DO COMMON SETUP
002116' 062502                          ADD     (R5)+,R2                ;ADD INCR VALUE TO SECT #
002120' 020203                  10$:    CMP     R2,R3                   ;SECT # IN RANGE?
002122' 103403                          BLO     15$                     ;N,Y-15$
002124' 160302                          SUB     R3,R2                   ;ADJ SECT # DOWNWARDS
002126' 005201                          INC     R1                      ;ADD 1 TO HEAD #
002130' 000773                          BR      10$                     ;GO CK NEW SECT # VALUE
002132' 020127  000024          15$:    CMP     R1,#20.                 ;HEAD # IN RANGE?
002136' 103404                          BLO     20$                     ;N,Y-20$
002140' 162701  000024                  SUB     #20.,R1                 ;ADJ HEAD # DOWNWARDS
002144' 005200                          INC     R0                      ;ADD 1 TO CYL #
002146' 000771                          BR      15$                     ;GO CHECK NEW HEAD #
002150' 020004                  20$:    CMP     R0,R4                   ;CYL # IN RANGE?
002152' 103402                          BLO     STEPEX                  ;N,Y-STEPEX
002154' 160400                          SUB     R4,R0                   ;ADJ CYL # DOWNWARDS
002156' 000774                          BR      20$                     ;GO CK IT NOW
002160' 010067  175620          STEPEX: MOV     R0,CYL                  ;STORE NEW CYL,HEAD,SECT VALUES
002164' 010167  175616                  MOV     R1,HEAD
002170' 010267  175614                  MOV     R2,SECT
002174' 000205                          RTS     R5                      ;EXIT TO USER PROG

002176' 016700  175602          STPCOM: MOV     CYL,R0                  ;GET CYL, HEAD, & SECT VALUES
002202' 016701  175600                  MOV     HEAD,R1                 ;IN REGISTERS
002206' 016702  175576                  MOV     SECT,R2
002212' 004767  003162                  JSR     PC,SUPTAD               ;GET PROG TBL ADR IN R3
002216' 012704  000313                  MOV     #203.,R4                ;PRESET # OF CYL'S TO RP02
002222' 032763  000020  000032          BIT     #20,PMDLCD(R3)          ;THIS AN RP03?
002230' 001402                          BEQ     STPC10                  ;Y,N-STPC10
002232' 012704  000626                  MOV     #406.,R4                ;SET UP # OF RP03 CYL'S
002236' 012703  000012          STPC10: MOV     #10.,R3                 ;SET UP # OF SECTORS
002242' 000207                          RTS     PC                      ;EXIT IN-LINE

                                        ;"STEPDN" FUNCTION ROUTINE
                                                ;JSR    R5,STEPDN       FUNCTION CALL
                                                ;.WORD  NBR             DECREMENT FACTOR
002244' 004767  177726          STEPDN: JSR     PC,STPCOM               ;DO COMMON SETUP
002250' 162502                          SUB     (R5)+,R2                ;SUB DECR FACTOR FROM SECT #
002252' 020203                  40$:    CMP     R2,R3                   ;SECT # IN RANGE?
002254' 103403                          BLO     45$                     ;N,Y-45$
002256' 060302                          ADD     R3,R2                   ;ADJ SECT # UPWARDS
002260' 005301                          DEC     R1                      ;DECR HEAD # BY 1
002262' 000773                          BR      40$                     ;GO CK NEW SECT # VALUE
002264' 020127  000024          45$:    CMP     R1,#20.                 ;HEAD # IN RANGE?
002270' 103404                          BLO     50$                     ;N,Y-50$
```

```
                                        ADD     #20.,R1                 ;ADJ IT UPWARDS
                                        DEC     R0                      ;DECR CYL # BY 1
                                        BR      45$                     ;GO CHECK NEW HEAD #
                                50$:    CMP     R0,R4                   ;CYL # IN RANGE?
                                        BLO     STEPEX                  ;N,Y-STEPEX
                                        ADD     R4,R0                   ;ADJ IT UPWARDS
                                        BR      50$                     ;GO CK IT NOW


                                        ;"WAIT" FUNCTION ROUTINE

                                                ;JSR    R5,WAIT         FUNCTION CALL

002312' 042767  100000  175462  WAIT:  BIC     #WAITMD,DFLGWD          ;RESET THE "NOWAIT" FLAG
002320' 004767  001552          JSR     PC,CKDBSY               ;WAIT IF BUSY & DO TERMINATION
002324' 004767  001774          JSR     PC,RINTV                ;RESET THE INTERRUPT VECTOR
002330' 000205                  RTS     R5                      ;EXIT IN-LINE


                                        ;"NOWAIT" FUNCTION ROUTINE

                                                ;JSR    R5,NOWAIT       FUNCTION CALL

002332' 052767  100000  175442  NOWAIT: BIS     #WAITMD,DFLGWD          ;SET THE "NOWAIT" FLAG
002340' 000205                  RTS     R5                      ;EXIT IN-LINE


                                        ;"HDRON" FUNCTION ROUTINE

                                                ;JSR    R5,HDRON        FUNCTION CALL

002342' 052767  014000  176766  HDRON:  BIS     #HEADER+MODE,RPCSV      ;SET THE HEADER & MODE BITS
002350' 000205                  RTS     R5                      ;EXIT IN-LINE


                                        ;"HDROFF" FUNCTION ROUTINE

                                                ;JSR    R5,HDROFF       FUNCTION CALL

002352' 042767  014000  176756  HDROFF: BIC     #HEADER+MODE,RPCSV      ;RESET THE HEADER & MODE BITS
002360' 000205                  RTS     R5                      ;EXIT IN-LINE


                                        ;"MODE11" FUNCTION ROUTINE

                                                ;JSR    R5,MODE11       FUNCTION CALL

002362' 042767  010000  176746  MODE11: BIC     #MODE,RPCSV             ;RESET THE MODE BIT
002370' 000205                  RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;"MODE10" FUNCTION ROUTINE
                                        ;
                                        ;JSR    R5,MODE10       FUNCTION CALL
                                        ;
002372' 052767  010000  176736  MODE10: BIS    #MODE,RPCSV     ;SET THE MODE BIT
002400' 000205                          RTS    R5              ;EXIT IN-LINE
                                        ;
                                        ;"IDLE" & "CRESET" FUNCTION ROUTINES
                                        ;
                                        ;JSR    R5,IDLE         FUNCTION CALL
                                        ;JSR    R5,CRESET       FUNCTION CALL
                                        ;
002402'                         IDLE:
002402' 004767  001470          CRESET: JSR    PC,CKDBSY       ;GO CK IF DEV IS BUSY
002406' 005267  176700                  INC    MISCNT          ;ADD 1 TO MISC. CMND CNT
002412' 005067  175404                  CLR    ERRI            ;RESET THE ERROR INDICATOR
002416' 012714  000001                  MOV    #1,(R4)         ;ISSUE THE IDLE FUNCTION CODE
002422' 012700  000012                  MOV    #10.,R0         ;SET UP DELAY COUNT
002426' 005300                  5$:     DEC    R0              ;DELAY FOR A FEW MICROSECONDS
002430' 001376                          BNE    5$
002432' 012714  000001                  MOV    #1,(R4)         ;ISSUE THE IDLE FUNCTION CODE
002436' 105714                  10$:    TSTB   (R4)            ;READY SET YET?
002440' 100407                          BMI    20$             ;N,Y-20$
002442' 005300                          DEC    R0              ;TIMED OUT?
002444' 100774                          BMI    10$             ;Y,N-10$
002446' 004567  001726                  JSR    R5,ERRCS        ;GO ISSUE CRESET TIMEOUT ERROR
002452' 002416                          .WORD  CRTO-ERMBAS
002454' 000177  175370                  JMP    @CUPGER         ;GO TO ERR RET POINT IN MPG
002460' 000205                  20$:    RTS    R5              ;EXIT IN-LINE TO USER'S PROG
```

```
                                .SBTTL  RP02/RP03 INTERRUPT TYPE I/O FUNCTION ROUTINES


                                ;"READ" FUNCTION ROUTINE

                                        ;JSR    R5,READ         FUNCTION CALL
                                        ;.WORD  ADR             DATA ADDRESS (BITS 16 - 17)
                                        ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                        ;.WORD  CNT             BYTE COUNT
                                        ;.WORD  DEV             (NOT USED)

002462' 012702  000105          READ:   MOV     #105,R2         ;SET UP READ FUNCT CODE
002466' 012701  000275                  MOV     #275,R1         ;SET UP CMND FLAG WORD
002472' 004767  001400          RDCOM:  JSR     PC,CKDBSY       ;GO CK IF DEV IS BUSY
002476' 005267  176600                  INC     RDCNT           ;ADD 1 TO READ CMND CNT
002502' 010700                          MOV     PC,R0           ;SET UP ADR OF BYTES READ CNT
002504' 062700  176564                  ADD     #BYRD+2-.,R0
002510' 000456                          BR      CMDCOM          ;GO TO CMND COMMON PROCESSING


                                ;"WRITE" FUNCTION ROUTINE

                                        ;JSR    R5,WRITE        FUNCTION CALL
                                        ;.WORD  ADR             DATA ADDRESS (BITS 16 - 17)
                                        ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                        ;.WORD  CNT             BYTE COUNT
                                        ;.WORD  DEV             (NOT USED)

002512' 012702  000103          WRITE:  MOV     #103,R2         ;SET UP WRITE FUNCT CODE
002516' 012701  000275                  MOV     #275,R1         ;SET UP CMND FLAG WORD
002522' 004767  001350          WRCOM:  JSR     PC,CKDBSY       ;GO CK IF DEV IS BUSY
002526' 005267  176552                  INC     WRCNT           ;ADD 1 TO WRITE CMND CNT
002532' 010700                          MOV     PC,R0           ;SET UP ADR OF BYTES WRITTEN CNT
002534' 062700  176540                  ADD     #BYWR+2-.,R0
002540' 000442                          BR      CMDCOM          ;GO TO CMND COMMON PROCESSING


                                ;"RDNOSK" FUNCTION ROUTINE

                                        ;JSR    R5,RDNOSK       FUNCTION CALL
                                        ;.WORD  ADR             DATA ADDRESS (BITS 16 - 17)
                                        ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                        ;.WORD  CNT             BYTE COUNT

002542' 012702  000117          RDNOSK: MOV     #117,R2         ;SET UP READ (NO SEEK) FUNCT CODE
002546' 012701  000256                  MOV     #256,R1         ;SET UP CMND FLAG WORD
002552' 000747                          BR      RDCOM           ;GO TO COMMON READ PROCESSING
```

```
                                        ;"WRNOSK" FUNCTION ROUTINE

                                                ;JSR    R5,WRNOSK       FUNCTION CALL
                                                ;.WORD  ADR             DATA ADDRESS (BITS 16 - 17)
                                                ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD  CNT             BYTE COUNT

002554' 012702  000113          WRNOSK: MOV     #113,R2                 ;SET UP WRITE (NO SEEK) FUNCT CODE
002560' 012701  000256                  MOV     #256,R1                 ;SET UP CMND FLAG WORD
002564' 000756                          BR      WRCOM                   ;GO TO WRITE COMMON PROCESSING


                                        ;"WRCK" FUNCTION ROUTINE

                                                ;JSR    R5,WRCK         FUNCTION CALL
                                                ;.WORD  ADR             DATA ADDRESS (BITS 16 - 17)
                                                ;.WORD  ADR             DATA ADDRESS (BITS 0 - 15)
                                                ;.WORD  CNT             BYTE COUNT

002566' 012702  000107          WRCK:   MOV     #107,R2                 ;SET UP WRITE CHECK FUNCT CODE
002572' 012701  000276                  MOV     #276,R1                 ;SET UP CMND FLAG WORD
002576' 004767  001274                  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
002602' 005267  176500                  INC     CKCNT                   ;ADD 1 TO CHECK CMND COUNT
002606' 010700                          MOV     PC,R0                   ;SET UP ADR OF BYTES
002610' 062700  176470                  ADD     #BYCK+2-.,R0            ;CHECKED COUNT
002614' 000414                          BR      CMDCOM                  ;GO TO CMND COM PROCESSING


                                        ;"SEEK" FUNCTION ROUTINE

                                                ;JSR    R5,SEEK         FUNCTION CALL

002616' 012702  000111          SEEK:   MOV     #111,R2                 ;SET UP SEEK FUNCT CODE
002622' 012701  000160          SKCOM:  MOV     #160,R1                 ;SET UP CMND FLAG WORD
002626' 004767  001244                  JSR     PC,CKDBSY               ;GO CK IF DEV IS BUSY
002632' 005267  176452                  INC     SKCNT                   ;ADD 1 TO SEEK CMND COUNT
002636' 000403                          BR      CMDCOM                  ;GO TO CMND COMMON PROCESSING


                                        ;"HOMESK" & "RECAL" FUNCTION ROUTINES

                                                ;JSR    R5,HOMESK       FUNCTION CALL
                                                ;JSR    R5,RECAL        FUNCTION CALL

002640'                         HOMESK:
002640' 012702  000115          RECAL:  MOV     #115,R2                 ;SET UP HOME SEEK FUNCT CODE
002644' 000766                          BR      SKCOM                   ;GO TO COMMON SEEK PROCESSING
```

```
                                ;INTERRUPT TYPE I/O FUNCTION COMMON PROCESSING ROUTINE

                                        ;R4 = ADR OF RPCS DEV REG
                                        ;R3 = PROG TBL ADR
                                        ;R2 = FUNCTION CODE
                                        ;R1 = COMMAND FLAG WORD
                                        ;R0 = ADR OF BYTE COUNT, IF APPLICABLE


                                        ;CMND FLAGWORD FORMAT:

                                                ;BIT 7 = 200 = PERFORM RETRIES ON CMND
                                                ;BIT 6 = 100 = TWO INTS & TERMINATES WITH ATTN
                                                ;BIT 5 = 040 = SET UP SECT #
                                                ;BIT 4 = 020 = SET UP CYL AND HD #
                                                ;BIT 3 = 010 = INCREMENT BYTE COUNTS
                                                ;BIT 2 = 004 = DATA TRANSFER CMND
                                                ;BIT 1 = 002 = 3 ARGUMENT CMND
                                                ;BIT 0 = 001 = 4 ARGUMENT CMND

002646' 010067  176372          CMDCOM: MOV     R0,CNTADR               ;SAVE ADR OF BYTE COUNT
002652' 010167  176370                  MOV     R1,CURFLG               ;SAVE FLAGWD FOR TERMINATION
002656' 032701  000003                  BIT     #3,R1                   ;THIS CMND HAVE BUS ADR & WD CNT?
002662' 001416                          BEQ     10$                     ;Y,N-10$
002664' 012567  176362                  MOV     (R5)+,CURADR            ;STORE 2 WORD BUS ADR
002670' 012567  176360                  MOV     (R5)+,CURADR+2
002674' 012500                          MOV     (R5)+,R0                ;GET BYTE COUNT
002676' 000241                          CLC                             ;MAKE IT A WORD COUNT
002700' 006000                          ROR     R0
002702' 005400                          NEG     R0                      ;MAKE IT NEGATIVE
002704' 010067  176346                  MOV     R0,CURCNT               ;SAVE IT
002710' 032701  003001                  BIT     #1,R1                   ;THERE A 4TH WORD?
002714' 001401                          BEQ     10$                     ;Y,N-10$
002716' 005725                          TST     (R5)+                   ;BYPASS IT
002720' 116300  000035          10$:    MOVB    PCURDV(R3),R0           ;GET CURRENT UNIT #
002724' 020027  000007                  CMP     R0,#7                   ;IS IT A VALID UNIT #?
002730' 101411                          BLOS    20$                     ;N,Y-20$
002732' 004567  001450                  JSR     R5,ERRCS1               ;ISSUE INV UNIT # ERROR MSG
002736' 002634                          .WORD   INVDVN-ERMBAS
002740' 005267  176352                  INC     DATAER                  ;ADJUST ERROR COUNTS
002744' 005367  176344                  DEC     ERRCNT
002750' 000177  175074          15$:    JMP     @CUPGER                 ;GO TO MPG ERR RETURN POINT
002754' 000300                  20$:    SWAB    R0                      ;ALIGN UNIT # BITS
002756' 050002                          BIS     R0,R2                   ;SET UNIT # INTO CMND CODE WORD
002760' 056702  176352                  BIS     RPCSV,R2                ;SET HEADER & MODE BITS IN CMND CODE
002764' 010267  176260                  MOV     R2,CURCMD               ;SAVE CURR CMND CODE
002770' 000300                          SWAB    R0                      ;MOVE UNIT # TO HIGH BYTE
002772' 110064  000001                  MOVB    R0,1(R4)                ;OF RPCS TO SELECT THE DRIVE
002776' 032764  040000  177774          BIT     #40000,RPDS(R4)         ;IS THE UNIT ON-LINE?
003004' 001004                          BNE     30$                     ;N,Y-30$
003006' 004567  001366                  JSR     R5,ERRCS                ;ISSUE UNIT OFF-LINE ERROR MSG
003012' 002601                          .WORD   UNOFFL-ERMBAS
003014' 000755                          BR      15$                     ;GO TO ERROR EXIT
003016' 005764  177774          30$:    TST     RPDS(R4)                ;IS UNIT READY?
003022' 100404                          BMI     40$                     ;N,Y-40$
```

```
 840 003024' 004567 001350                  JSR     R5,ERRCS        ;ISSUE UNIT NOT RDY ERROR MSG
 841 003030' 002617                         .WORD   UNWRDY-ERMBAS
 842 003032' 000746                         BR      15$             ;GO TO ERROR EXIT
 843 003034' 032764 001000  177774  40$:    BIT     #1000,RPDS(R4)  ;IS THE UNIT UNSAFE?
 844 003042' 001404                         BEQ     50$             ;Y,N-50$
 845 003044' 004567 001330                  JSR     R5,ERRCS        ;ISSUE UNSAFE ON INIT ERROR MSG
 846 003050' 002460                         .WORD   INITUS-ERMBAS
 847 003052' 000736                         BR      15$             ;GO TO ERROR EXIT
 848 003054' 006300                 50$:    ASL     R0              ;USING THE UNIT #, INDEX INTO
 849 003056' 060700                         ADD     PC,R0           ;THE ATTN TBL FOR BITS FOR
 850 003060' 062700 000136                  ADD     #ATATBL-.,R0    ;THIS UNIT #
 851 003064' 112067 000146                  MOVB    (R0)+,MYATA     ;STORE UNIT'S ATTN BIT
 852 003070' 111067 000144                  MOVB    (R0),OTHATA     ;STORE OTHER ATTN BITS
 853 003074' 016400 177774                  MOV     RPDS(R4),R0     ;GET THE ATTN BITS
 854 003100' 136700 000132                  BITB    MYATA,R0        ;IS MY ATTN ALREADY SET?
 855 003104' 001403                         BEQ     55$             ;Y,N-55$
 856 003106' 116764 000124  177774          MOVB    MYATA,RPDS(R4)  ;KNOCK IT DOWN
 857 003114' 004767 000122          55$:    JSR     PC,SUIORG       ;GO SET UP REGS FOR I/O
 858 003120' 016767 174666  176134          MOV     RTRY,CURRTY     ;INITIALIZE RETRY COUNT
 859 003126' 005067 176132                  CLR     RTRYIP          ;CLEAR RETRY IN PROGRESS FLAG
 860 003132' 012767 002504  001132          MOV     #IOTERM-ERMBAS,INTEAD ;INIT TERMINATION ERROR MSG
 861 003140' 042767 000005  174634          BIC     #IOERR+W4ATTN,DFLGWD ;MSKP INT FLAGS
 862 003146' 052767 000002  174626          BIS     #DOTERM,DFLGWD  ;SET THE "PROCESS TERMINATION" FLAG
 863 003154' 052713 000010                  BIS     #WT4IOT,(R3)    ;SET WAITING FOR I/O TERM FLAG
 864 003160' 116764 176065  000001          MOVB    CURCMD+1,1(R4)  ;LOAD HIGH BYTE OF RPCS REG
 865 003166' 110214                         MOVB    R2,(R4)         ;ISSUE SPECIFIED CMND
 866 003170' 005767 174606                  TST     DFLGWD          ;"NOWAIT" BIT SET?
 867 003174' 100003                         BPL     60$             ;Y,N-60$
 868 003176' 042713 000010                  BIC     #WT4IOT,(R3)    ;RESET WAITING FOR I/O TERM
 869 003202' 000404                         BR      70$             ;GO TO EXIT
 870 003204' 004577 174636          60$:    JSR     R5,@CIOBSY      ;WAIT FOR I/O TO COMPLETE
 871 003210' 004767 000762                  JSR     PC,PROCTM       ;GO PROCESS TERMINATION
 872 003214' 000205                 70$:    RTS     R5              ;EXIT IN-LINE TO USER PROG
 873
 874
 875 003216'     001     376        ATATBL: .BYTE   001,376
 876 003220'     002     375                .BYTE   002,375
 877 003222'     004     373                .BYTE   004,373
 878 003224'     010     367                .BYTE   010,367
 879 003226'     020     357                .BYTE   020,357
 880 003230'     040     337                .BYTE   040,337
 881 003232'     100     277                .BYTE   100,277
 882 003234'     200     177                .BYTE   200,177
 883
 884 003236' 000000         MYATA:  .WORD   0
 885 003240' 000000         OTHATA: .WORD   0
```

```
                                        ;SET UP DEVICE REGS FOR I/O
                                        ;
                                        ;JSR    PC,SUIORG       S/R CALL
                                        ;
                                        ;R4 = RPCS ADR
                                        ;R3 = PROG TBL ADR
                                        ;R2 = CMND CODE
                                        ;R1 = CMND FLAGWORD
                                        ;
                                        ;DESTROYS R0
                                        ;
003242' 032701  000060          SUIORG: BIT     #60,R1                  ;NEED TO SET UP CYL/HEAD/SECT?
003246' 001414                          BEQ     10$                     ;Y,N-10$
003250' 116764  174534  000010          MOVB    SECT,RPDA(R4)           ;STORE SECT #
003256' 032701  000020                  BIT     #20,R1                  ;NEED CYL & HEAD #?
003262' 001406                          BEQ     10$                     ;Y,N-10$
003264' 116764  174516  000011          MOVB    HEAD,RPDA+1(R4)         ;LOAD HEAD #
003272' 016764  174506  000006          MOV     CYL,RPCA(R4)            ;LOAD CYL #
003300' 032701  000004          10$:    BIT     #4,R1                   ;DATA XFER CMND?
003304' 001417                          BEQ     20$                     ;Y,N-20$
003306' 016700  175740                  MOV     CURADR,R0               ;GET HIGH BITS OF ADR
003312' 042700  177774                  BIC     #177774,R0              ;RESET BITS ABOVE A17
003316' 006300                          ASL     R0                      ;ALIGN BITS A16 & A17
003320' 006300                          ASL     R0                      ;TO THEIR POSITIONS IN RPCS
003322' 006300                          ASL     R0
003324' 006300                          ASL     R0
003326' 050002                          BIS     R0,R2                   ;SET THEM INTO CMND CODE WORD
003330' 016764  175720  000004          MOV     CURADR+2,RPBA(R4)       ;LOAD BITS 0-15 OF ADR
003336' 016764  175714  000002          MOV     CURCNT,RPWC(R4)         ;LOAD WORD COUNT
003344' 012763  072460  000030  20$:    MOV     #30000.,PTOCNT(R3)      ;INITIALIZE 10 SEC TIMEOUT CNT
003352' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                        .SBTTL  RP02/RP03 INTERRUPT SERVICE ROUTINE
                                
003354' 004067 001766           RPINT:  JSR     R0,SAVREG               ;SAVE ALL REGISTERS
003360' 004567 002040                   JSR     R5,STSTAT               ;GO STORE ALL DEV REG'S
003364' 176602                          .WORD   ISTAT-.
003366' 005267 175742                   INC     INTCNT                  ;ADD 1 TO INTERRUPT CNT
003372' 004767 002002                   JSR     PC,SUPTAD               ;SET UP PROG TBL & RPCS1 ADR'S
003376' 016701 175644                   MOV     CURFLG,R1               ;GET THIS CMND'S FLGWD
003402' 032767 000004 174372            BIT     #W4ATTN,DFLGWD          ;WAITING FOR ATTN INT?
003410' 001032                          BNE     ATNINT                  ;N,Y-ATNINT
003412' 042714 000100                   BIC     #100,(R4)               ;RESET INTERRUPT ENABLE
003416' 016467 000002 175634            MOV     RPWC(R4),FINCNT         ;STORE FINAL WORD COUNT
003424' 032714 140000                   BIT     #140000,(R4)            ;'ERR' OR 'HE' SET?
003430' 001055                          BNE     ERRFND                  ;N,Y-ERRFND
003432' 032701 000100                   BIT     #100,R1                 ;CMND RESULT IN TWO INTERRUPTS?
003436' 001411                          BEQ     CLRWTF                  ;Y,N-CLRWTF
003440' 052767 000004 174334            BIS     #W4ATTN,DFLGWD          ;SET 'WAITING FOR ATTN' FLAG
003446' 052714 020000                   BIS     #20000,(R4)             ;SET ATTN INTERRUPT ENABLE
003452' 000405                          BR      INTEX                   ;GO TO INTERRUPT EXIT
003454' 052767 000001 174320    SETERR: BIS     #IOERR,DFLGWD           ;SET THE TERMINATION I/O ERR FLAG
003462' 042713 000010           CLRWTF: BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
003466' 004067 001670           INTEX:  JSR     R0,RESREG               ;RESTORE ALL REGISTERS
003472' 000177 174376                   JMP     @RTNINT                 ;EXIT FROM INTERRUPT

003476' 042714 020000           ATNINT: BIC     #20000,(R4)             ;RESET ATTN INT ENABLE
003502' 032714 140000                   BIT     #140000,(R4)            ;'ERR' OR 'HE' SET?
003506' 001362                          BNE     SETERR                  ;N,Y-SETERR
003510' 136764 177522 177774            BITB    MYATA,RPDS(R4)          ;IS MY ATTN BIT SET?
003516' 001404                          BEQ     10$                     ;Y,N-10$
003520' 116764 177512 177774            MOVB    MYATA,RPDS(R4)          ;RESET MY ATTN BIT
003526' 000755                          BR      CLRWTF                  ;GO TO INT EXIT
003530' 136764 177504 177774    10$:    BITB    OTHATA,RPDS(R4)         ;IS ANOTHER ATTN BIT SET?
003536' 001406                          BEQ     20$                     ;Y,N-20$
003540' 116764 177474 177774            MOVB    OTHATA,RPDS(R4)         ;RESET OTHER ATTN BITS
003546' 052714 020000                   BIS     #20000,(R4)             ;SET ATTN INT ENABLE AGAIN
003552' 000745                          BR      INTEX                   ;GO TO INT EXIT
003554' 012767 002532 000510    20$:    MOV     #NOATA-ERMBAS,INTEAD    ;SET UP 'ATTN NOT SET 'ERR MSG ADR
003562' 000734                          BR      SETERR                  ;GO SET ERROR FLAG & EXIT

003564' 032764 177407 177776    ERRFND: BIT     #177407,RPER(R4)        ;ANY HARD ERROR BITS IN RPER?
003572' 001330                          BNE     SETERR                  ;N,Y-SETERR
003574' 016400 177774                   MOV     RPDS(R4),R0             ;GET THE RPDS REG
003600' 032700 015000                   BIT     #15000,R0               ;ANY HARD ERROR BITS IN RPDS?
003604' 001323                          BNE     SETERR                  ;N,Y-SETERR
003606' 005100                          COM     R0                      ;FLIP RPDS BITS
003610' 032700 140000                   BIT     #140000,R0              ;STILL ON-LINE & READY?
003614' 001317                          BNE     SETERR                  ;Y,N-SETERR
003616' 016400 177776                   MOV     RPER(R4),R0             ;GET THE RPER REG
003622' 032700 000370                   BIT     #370,R0                 ;RETRYABLE ERROR?
003626' 001712                          BEQ     SETERR                  ;Y,N-SETERR
003630' 032701 000200                   BIT     #200,R1                 ;DO RETRIES ON THIS CMND?
003634' 001707                          BEQ     SETERR                  ;Y,N-SETERR

003636' 005767 175422           CKRTRY: TST     RTRYIP                  ;ALREADY DONE RETRIES ON THIS CMND?
003642' 001070                          BNE     55$                     ;N,Y-55$
```

```
 975  003644' 005767 175412           TST   CURRTY          ;ARE RETRIES SPECIFIED?
 976  003650' 001701                  BEQ   SETERR          ;Y,N-SETERR
 977  003652' 032700 000020           BIT   #20,R0          ;TIMING ERROR?
 978  003656' 001411                  BEQ   42$             ;Y,N-42$
 979  003660' 005267 175434           INC   TIECNT          ;ADD 1 TO TIMING ERROR COUNT
 980  003664' 012767 046524 003434    MOV   #"TM,RTYID      ;SET RETRY I.D. TO "TMEE"
 981  003672' 012767 042505 003430    MOV   #"EE,RTYID+2
 982  003700' 000451                  BR    55$             ;GO CK RETRY COUNT
 983  003702' 032700 000040     42$:  BIT   #40,R0          ;CHECKSUM ERROR?
 984  003706' 001411                  BEQ   44$             ;Y,N-44$
 985  003710' 005267 175406           INC   CSECNT          ;ADD 1 TO CHECKSUM ERROR COUNT
 986  003714' 012767 051503 003404    MOV   #"CS,RTYID      ;SET RETRY I.D. TO "CSME"
 987  003722' 012767 042515 003400    MOV   #"ME,RTYID+2
 988  003730' 000435                  BR    55$             ;GO CK RETRY COUNT
 989  003732' 032700 000100     44$:  BIT   #100,R0         ;WORD PARITY ERROR?
 990  003736' 001406                  BEQ   46$             ;Y,N-46$
 991  003740' 005267 175360           INC   WPECNT          ;ADD 1 TO WORD PARITY ERROR COUNT
 992  003744' 012767 050127 003354    MOV   #"WP,RTYID      ;SET RETRY I.D. TO "WPE"
 993  003752' 000421                  BR    50$             ;GO CK RETRY COUNT
 994  003754' 032700 000200     46$:  BIT   #200,R0         ;LONGITUDINAL PARITY ERROR?
 995  003760' 001406                  BEQ   48$             ;Y,N-48$
 996  003762' 005267 175340           INC   LPECNT          ;ADD 1 TO LONGITUDINAL PARITY COUNT
 997  003766' 012767 050114 003332    MOV   #"LP,RTYID      ;SET RETRY I.D. TO "LPE"
 998  003774' 000410                  BR    50$             ;GO CK RETRY COUNT
 999  003776' 032700 000010     48$:  BIT   #10,R0          ;WRITE CHECK ERROR?
1000  004002' 001410                  BEQ   55$             ;Y,N-55$
1001  004004' 005267 175320           INC   WCECNT          ;ADD 1 TO WRITE CHECK ERROR COUNT
1002  004010' 012767 041527 003310    MOV   #"WC,RTYID      ;SET RETRY I.D. TO "WCE"
1003  004016' 012767 020105 003304 50$:  MOV   #"E ,RTYID+2
1004  004024' 005367 175232     55$:  DEC   CURRTY          ;DECREMENT RETRY COUNT
1005  004030' 100004                  BPL   60$             ;CNT EXHAUSTED? (Y,N-60$)
1006  004032' 012767 002547 000232    MOV   #RTYEXH-ERMBAS,INTEAD ;SET UP EXHAUSTED RETRIES ERR MSG ADR
1007  004040' 000605                  BR    SETERR          ;GO TO ERROR EXIT
1008  004042' 005267 175264     60$:  INC   RETRYS          ;ADD 1 TO RETRY TOTAL CNT
1009  004046' 005267 175212           INC   RTRYIP          ;SET RETRY IN PROGRESS FLAG
1010  004052' 016702 175172           MOV   CURCMD,R2       ;GET CURR CMND IN R2
1011  004056' 004767 177160           JSR   PC,SUIORG       ;SET UP DEV REGS
1012  004062' 116764 175163 000001    MOVB  CURCMD+1,1(R4)  ;LOAD HIGH BYTE OF RPCS
1013  004070' 110214                  MOVB  R2,(R4)         ;RE-ISSUE THE ORIG CMND
1014  004072' 000167 177370           JMP   INTEX           ;GO TO INT EXIT
```

```
                                        .SBTTL  SUBROUTINES FOR RP02/RP03 FUNCTION ROUTINES


                                        ;CHECK IF DEVICE IS BUSY AND WAIT IF IT IS

                                                ;JSR    PC,CKDBSY       S/R CALL
                                                ;
                                                ;DESTROYS R0,R3,R4
                                                ;
                                                ;ON EXIT:  R3 = PROG TBL ADR
                                                ;          R4 = RPCS1 ADR

004076' 004767  001276          CKDBSY: JSR     PC,SUPTAD               ;SET UP PROG TBL & RPCS1 ADR'S
004102' 032714  020100          10$:    BIT     #20100,(R4)             ;EITHER INT ENABLE ON?
004106' 001403                          BEQ     20$                     ;Y,N-20$
004110' 004577  173732                  JSR     R5,@CIOBSY              ;RELEASE CONTROL
004114' 000772                          BR      10$                     ;GO CK AGAIN
004116' 032767  000002  173656  20$:    BIT     #DOTERM,DFLGWD          ;HAVE TO PROCESS PREV TERMINATION?
004124' 001403                          BEQ     30$                     ;Y,N-30$
004126' 004767  000044                  JSR     PC,PROCTM               ;GO PROCESS TERMINATION
004132' 000763                          BR      10$                     ;GO CK INT ENABLE AGAIN
004134' 016767  173666  000012  30$:    MOV     IVCTAD,40$              ;STORE INT VECTOR ADR
004142' 016767  173662  000006          MOV     PSWD,45$                ;STORE PROC STATUS WORD
004150' 004577  173712                  JSR     R5,@SETVEC              ;GO SET UP INTERRUPT VECTOR
004154' 000000                  40$:    .WORD   XXXX                    ;  INT VECTOR ADR
004156' 000000                  45$:    .WORD   XXXX                    ;  PSW
004160' 177174                          .WORD   RPINT-.                 ;  REL INT ROUT ADR
004162' 010567  175054          STMADR: MOV     R5,OBJADR               ;SAVE CURR USER STMNT ADR
004166' 162767  000004  175046          SUB     #4,OBJADR
004174' 000207                          RTS     PC                      ;EXIT IN-LINE


                                        ;PROCESS TERMINATION OF PREVIOUS I/O FUNCTION

                                                ;JSR    PC,PROCTM       S/R CALL
                                                ;
                                                ;R3 = PROG TABLE ADR
                                                ;
                                                ;DESTROYS R0

004176' 010146                  PROCTM: MOV     R1,-(SP)                ;SAVE R1 & R2
004200' 010246                          MOV     R2,-(SP)
004202' 042767  000002  173572          BIC     #DOTERM,DFLGWD          ;RESET PROCESS TERMINATION FLAG
004210' 032767  000010  175030          BIT     #10,CURFLG              ;INCR BYTE COUNT?
004216' 001417                          BEQ     20$                     ;Y,N-20$
004220' 016700  175032                  MOV     CURCNT,R0               ;GET INITIAL WORD CNT
004224' 005400                          NEG     R0                      ;MAKE IT POSITIVE AGAIN
004226' 016701  175026                  MOV     FINCNT,R1               ;GET FINAL WORD CNT
004232' 100001                          BPL     10$                     ;IS IT NEGATIVE? (Y,N-10$)
004234' 005401                          NEG     R1                      ;MAKE IT POSITIVE
004236' 160100                  10$:    SUB     R1,R0                   ;SUB REMAINING CNT FROM INITIAL CNT
004240' 006300                          ASL     R0                      ;MAKE IT A BYTE CNT
004242' 010067  173552                  MOV     R0,SIZE                 ;STORE # OF BYTES ACTUALLY XFERRED
004246' 016701  174772                  MOV     CNTADR,R1               ;GET ADR OF BYTE CNT TOTALS
004252' 060011                          ADD     R0,(R1)                 ;ADD IN THIS CNT
004254' 005541                          ADC     -(R1)                   ;UPDATE MOST SIGNF WORD OF CNT
```

```
1072  004256' 032767 000001 173516 20$:    BIT     #IOERR,DFLGWD           ;WAS THERE AN ERROR?
1073  004264' 001412                       BEQ     PROCEX                  ;Y,N-PROCEX
1074  004266' 004567 000132                JSR     R5,ERRIS                ;GO ISSUE I/O TERMINATION
1075  004272' 002504                INTEAD: .WORD   IOTERM-ERMBAS           ;ERROR MSG
1076  004274' 004767 000024                JSR     PC,RINTV                ;RESET THE INT VECTOR
1077  004300' 012602                       MOV     (SP)+,R2                ;RESTORE R1 & R2
1078  004302' 012601                       MOV     (SP)+,R1
1079  004304' 004577 173540               JSR     R5,@CUPGER              ;GO TO MPG ERR RETN POINT
1080  004310' 000207                       RTS     PC                      ;RETURN IN-LINE
1081  004312' 004767 000006        PROCEX: JSR     PC,RINTV                ;GO RESET INT VECTOR
1082  004316' 012602                       MOV     (SP)+,R2                ;RESTORE R1 & R2
1083  004320' 012601                       MOV     (SP)+,R1
1084  004322' 000207                       RTS     PC                      ;EXIT IN-LINE
1085
1086
1087                                ;RESET INTERRUPT VECTOR S/R
1088
1089                                        ;JSR    PC,RINTV        S/R CALL
1090                                        ;
1091                                        ;R3 = PROG TBL ADR
1092                                        ;
1093                                        ;DESTROYS R0
1094
1095  004324' 004567 000020        RINTV:  JSR     R5,TVECT                ;GO CK IF I HAVE VECTOR CONTROL
1096  004330' 000406                       BR      RINTEX                  ;BR IF I DON'T
1097  004332' 016767 173470 000004        MOV     IVCTAD,10$              ;GET CURR INT VECT ADR
1098  004340' 004577 173524               JSR     R5,@CLRVEC              ;GO HAVE MPG CLEAR IT
1099  004344' 000000                10$:    .WORD   XXXX
1100  004346' 000207                RINTEX: RTS     PC                      ;EXIT IN-LINE
1101
1102
1103                                ;TEST INTERRUPT VECTOR S/R
1104
1105                                        ;JSR    R5,TVECT        S/R CALL
1106                                        ;BR     LABEL           EXECUTED IF NOT SAME
1107                                        ;
1108                                        ;R3 = PROG TBL ADR
1109                                        ;
1110                                        ;DESTROYS R0
1111
1112  004350' 016767 173452 000010 TVECT:  MOV     IVCTAD,20$              ;GET CURR INT VECT ADR
1113  004356' 016346 000004                MOV     PFWADR(R3),-(SP)        ;STORE FLGWD ADR TO IDENTIFY ME
1114  004362' 004577 173504                JSR     R5,@TSTVEC              ;DO I HAVE VECTOR CONTROL?
1115  004366' 000000                20$:    .WORD   XXXX                    ;   MPG WILL TELL ME SINCE I CAN'T
1116  004370' 176764                       .WORD   RPINT-.                 ;   GET AT LOWER MEM IF MEM MGMNT
1117  004372' 000401                       BR      TVECTX                  ;BR IF I DONT'T HAVE CNTRL
1118  004374' 005725                       TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
1119  004376' 000205                TVECTX: RTS     R5                      ;EXIT IN-LINE
```

```
                                        ;ERROR INFORMATION DISPLAY S/R
                                        ;
                                        ;JSR    R5,ERRCS        S/R CALL FOR CURR STATUS
                                        ;         ,ERRCS1       S/R CALL FOR CURR STATUS W/O STORING
                                        ;         ,ERRIS        S/R CALL FOR INT STATUS
                                        ;.WORD  MSGADR-ERMBAS   REL ADR OF ERROR MSG
                                        ;
                                        ;R3 = PROG TABLE ADR
                                        ;DESTROYS R0,R1,R2
                                        ;
004400' 004567 001020           ERRCS:  JSR     R5,STSTAT       ;STORE CURR STATUS
004404' 174610                          .WORD   CSTAT-.
004406' 012767 174246 000332    ERRCS1: MOV     #CSTAT-ERSTAD,ERSTAD    ;STORE ADR OF CURR STATUS
004414' 012767 174436 000144            MOV     #CSTAT+6-EBSBAS,EBSTAT
004422' 000406                          BR      ERRCOM          ;GO TO COMMON POINT
004424' 012767 174220 000314    ERRIS:  MOV     #ISTAT-ERSTAD,ERSTAD    ;STORE ADR OF LAST INT STATUS
004432' 012767 174410 000126            MOV     #ISTAT+6-EBSBAS,EBSTAT
004440' 012567 000066           ERRCOM: MOV     (R5)+,ERMBAS    ;STORE MSG ADR
004444' 005267 174644                   INC     ERRCNT          ;ADD 1 TO ERROR CNT
004450' 012767 000001 173344            MOV     #1,ERRI         ;SET THE ERROR INDICATOR
004456' 032763 020400 000002            BIT     #DOERCK+PRONER,POPSW(R3) ;ERROR CHECKING OR PRINTING INHIBITED?
004464' 001163                          BNE     ERREX           ;Y,N-ERREX
004466' 010446                          MOV     R4,-(SP)        ;SAVE R4 & R5
004470' 010546                          MOV     R5,-(SP)
004472' 005004                          CLR     R4              ;SET USER MODE PRINT FLAG
004474' 004767 000750                   JSR     PC,DEVID        ;DISPLAY DEVICE I.D.
004500' 010700                          MOV     PC,R0           ;GET START ADR OF ERROR MSG
004502' 062700 000030                   ADD     #ERMBAS-.,R0
004506' 061000                          ADD     (R0),R0
004510' 012701 177777                   MOV     #-1,R1          ;INITIALIZE MSG LENGTH
004514' 005201                  10$:    INC     R1              ;ADD 1 TO MSG LENGTH
004516' 105720                          TSTB    (R0)+           ;MSG TERMINATOR?
004520' 001375                          BNE     10$             ;Y,N-10$
004522' 010167 000006                   MOV     R1,ERMBAS+2     ;STORE MSG LENGTH
004526' 004567 001266                   JSR     R5,PRINT        ;PRINT ERROR MSG SPECIFIED
004532' 000000                  ERMBAS: .WORD   XXXX
004534' 000000                          .WORD   XXXX
004536' 026727 177770 002634            CMP     ERMBAS,#INVDVN-ERMBAS   ;INVALID UNIT # MSG OR HIGHER?
004544' 103103                          BHIS    ERRSNM          ;N,Y-ERRSNM
004546' 010701                          MOV     PC,R1           ;GET ADR OF CODE  AREA IN ERR MSG
004550' 062701 002300                   ADD     #CODFLD-.,R1
004554' 010700                          MOV     PC,R0           ;SET UP ADR OF ERROR CODE TBL
004556' 062700 000260                   ADD     #ERCDTB-.,R0
004562' 010702                          MOV     PC,R2           ;SET UP ADR OF STORED DEV REG'S
004564' 062702                  EBSBAS: ADD     (PC)+,R2
004566' 174436                  EBSTAT: .WORD   CSTAT+6-EBSBAS
004570' 012767 000015 000142            MOV     #13.,70$        ;INITIALIZE MSG LENGTH
004576' 012746 000100                   MOV     #64.,-(SP)      ;INITIALIZE CODE FIELD CNT
004602' 014205                  15$:    MOV     -(R2),R5        ;GET NEXT DEV REG WORD
004604' 000305                  17$:    SWAB    R5              ;GET DESIRED BYTE IN LOW BYTE
004606' 112004                  20$:    MOVB    (R0)+,R4        ;GET FLAG & LENGTH BYTE
004610' 001444                          BEQ     60$             ;END OF CODE TBL? (N,Y-60$)
004612' 122704 000377                   CMPB    #377,R4         ;GO TO NXT DEV REG WORD?
004616' 001771                          BEQ     15$             ;N,Y-15$
004620' 122704 000376                   CMPB    #376,R4         ;GO TO NXT BYTE IN DEV REG WORD?
004624' 001767                          BEQ     17$             ;N,Y-17$
```

```
1177  004626' 105704                    TSTB    R4              ;BIT VALUE OF 0 = AN ERROR CONDITION?
1178  004630' 100003                    BPL     30$             ;Y,N-30$
1179  004632' 131005                    BITB    (R0),R5         ;THIS BIT RESET IN DEV REG BYTE?
1180  004634' 001407                    BEQ     40$             ;N,Y-40$
1181  004636' 000402                    BR      35$             ;GO TO NXT TBL ENTRY
1182  004640' 131005            30$:    BITB    (R0),R5         ;THIS ERROR BIT SET IN DEV REG BYTE?
1183  004642' 001004                    BNE     40$             ;N,Y-40$
1184  004644' 042704 177770     35$:    BIC     #177770,R4      ;ISOLATE ENTRY LENGTH
1185  004650' 060400                    ADD     R4,R0           ;POINT AT NXT CODE TBL ENTRY
1186  004652' 000755                    BR      20$             ;GO CK FOR NXT CODE
1187  004654' 042704 177770     40$:    BIC     #177770,R4      ;ISOLATE I.D. NAME LENGTH + 1
1188  004660' 020416                    CMP     R4,(SP)         ;ENOUGH ROOM FOR NAME?
1189  004662' 101017                    BHI     60$             ;Y,N-60$
1190  004664' 060467 000050             ADD     R4,70$          ;ADJ MSG LENGTH FOR NAME
1191  004670' 005304                    DEC     R4              ;ADJ FOR BIT MASK CHAR
1192  004672' 005200                    INC     R0              ;POINT PAST BIT MASK
1193  004674' 021627 000100             CMP     (SP),#64.       ;FIRST ERROR CODE IN MSG?
1194  004700' 001403                    BEQ     50$             ;N,Y-50$
1195  004702' 112721 000054             MOVB    #',,(R1)+       ;MOVE COMMA TO MSG
1196  004706' 005316                    DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
1197  004710' 112021            50$:    MOVB    (R0)+,(R1)+     ;MOVE ERROR CODE TO MSG
1198  004712' 005316                    DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
1199  004714' 005304                    DEC     R4              ;MOVED ALL NAME CHARS?
1200  004716' 001374                    BNE     50$             ;Y,N-50$
1201  004720' 000732                    BR      20$             ;GO CK FOR MORE ERROR BITS
1202  004722' 005004            60$:    CLR     R4              ;SET USER MODE PRINT
1203  004724' 022627 000100             CMP     (SP)+,#64.      ;ANY ERROR CODES PUT IN MSG?
1204  004730' 001404                    BEQ     80$             ;Y,N-80$
1205  004732' 004567 001062             JSR     R5,PRINT        ;GO ISSUE ERROR BITS MSG
1206  004736' 002074                    .WORD   ERRMSG-.
1207  004740' 000116            70$:    .WORD   78.
1208  004742' 004567 000616     80$:    JSR     R5,DISPST       ;DISPLAY DEVICE REG'S
1209  004746' 000000            ERSTAD: .WORD   XXXX
1210  004750' 004767 000774             JSR     PC,PRTIWD       ;DISPLAY CYL,HEAD,SECT VALUES
1211  004754' 016300 000022     ERRSNM: MOV     PSRCST(R3),R0   ;GET ADR OF SRC STMNTS
1212  004760' 111001            110$:   MOVB    (R0),R1         ;SAVE STMNT LENGTH
1213  004762' 026067 000004 174252     CMP     4(R0),OBJADR    ;ERROR OCCUR ON THIS STMNT?
1214  004770' 001402                    BEQ     120$            ;N,Y-120$
1215  004772' 060100                    ADD     R1,R0           ;POINT AT NXT STMNT
1216  004774' 000771                    BR      110$            ;GO CK NXT STMNT
1217  004776' 005720            120$:   TST     (R0)+           ;SET UP ADR OF STMNT # DATA
1218  005000' 010701                    MOV     PC,R1           ;SET UP DATA OUTPUT ADR
1219  005002' 062701 002022             ADD     #STMNUM-.,R1
1220  005006' 004577 173050             JSR     R5,@DECASC      ;CONVERT IT TO ASCII
1221  005012' 012767 020040 002010      MOV     #20040,STMNUM+4 ;SET 2 LOW DIGITS TO SPACES
1222  005020' 004567 000774             JSR     R5,PRINT        ;ISSUE STMNT # MSG
1223  005024' 001770                    .WORD   STMNMG-.
1224  005026' 177762                    .WORD   -14.
1225  005030' 012605                    MOV     (SP)+,R5        ;RESTORE R5 & R4
1226  005032' 012604                    MOV     (SP)+,R4
1227  005034' 000205            ERREX:  RTS     R5              ;EXIT IN-LINE
```

```
                                        ;ERROR MESSAGE CODE TABLE
                                        ;
                                        ;377 = GO TO NEXT DEVICE REGISTER WORD
                                        ;376 = GO TO NEXT DEVICE REGISTER BYTE
                                        ;
                                        ;BYTE 0 CONTAINS FLAG BIT & I.D. NAME LENGTH
                                        ;
                                        ;       BITS 0-2 = LENGTH OF BIT MASK + I.D. NAME
                                        ;       BIT 7 = BIT = 0 IS AN ERROR CONDITION
                                        ;
                                        ;BYTE 1 IS THE BIT MASK
                                        ;
                                        ;BYTES 2 THRU ? ARE THE BIT'S ASCII I.D.
                                        ;
005036' 100004 051105    122  ERCDTB:  .ASCII  <004><200>/ERR/          ;RPCS - BYTE 1
005043'    003 044100    105           .ASCII  <003><100>/ME/
005047'    377                         .BYTE   377
005050' 100004 050127    126           .ASCII  <004><200>/WPV/          ;RPER - BYTE 1
005055'    004 043100 053125           .ASCII  <004><100>/FUV/
005062' 020004 054116    103           .ASCII  <004><040>/NXC/
005067'    004 047020 052130           .ASCII  <004><020>/NXT/
005074' 004004 054116    123           .ASCII  <004><010>/NXS/
005101'    006 050004 047522           .ASCII  <006><004>/PROGE/
005106' 042507
005110' 001005 046506 042524           .ASCII  <005><002>/FMTE/
005116' 000406 047515 042504           .ASCII  <006><001>/MODEE/
005124'    105
005125'    376                         .BYTE   376
005126' 100004 050114    105           .ASCII  <004><200>/LPE/          ;RPER - BYTE 0
005133'    004 053500 042520           .ASCII  <004><100>/WPE/
005140' 020005 051503 042515           .ASCII  <005><040>/CSME/
005146' 010006 044524 042515           .ASCII  <006><020>/TIMEE/
005154'    105
005155'    004 053410 042503           .ASCII  <004><010>/WCE/
005162' 002005 054116 042515           .ASCII  <005><004>/NXME/
005170' 001005 047505 042520           .ASCII  <005><002>/EOPE/
005176' 000407 051504 042513           .ASCII  <007><001>/DSKERR/
005204' 051122
005206'    377                         .BYTE   377
005207'    206 051600 051125           .ASCII  <206><200>/SURDY/        ;RPDS - BYTE 1
005214' 054504
005216' 040205 052523 046117           .ASCII  <205><100>/SUOL/
005224' 010004 047110    106           .ASCII  <004><020>/WNF/
005231'    005 051410 051525           .ASCII  <005><010>/SUSI/
005236'    111
005237'    005 051402 043125           .ASCII  <005><002>/SUFU/
005244'    125
005245'    005 051401 053525           .ASCII  <005><001>/SUWP/
005252'    120
005253'    376                         .BYTE   376
005254' 100006 052101 047124           .ASCII  <006><200>/ATTN7/        ;RPDS - BYTE 0
005262'    067
005263'    006 040500 052124           .ASCII  <006><100>/ATTN6/
005270' 033116
005272' 020006 052101 047124           .ASCII  <006><040>/ATTN5/
005300'    065
```

```
1274  005301'  006 040420 052124          .ASCII  <006><020>/ATTN4/
      005306'  032116
1275  005310'  004006 052101 047124          .ASCII  <006><010>/ATTN3/
      005316'  063
1276  005317'  006 040404 052124          .ASCII  <006><004>/ATTN2/
      005324'  031116
1277  005326'  001006 052101 047124          .ASCII  <006><002>/ATTN1/
      005334'  061
1278  005335'  006 040401 052124          .ASCII  <006><001>/ATTN0/
      005342'  030116
1279  005344'  000                        .BYTE   0                       ;TABLE TERMINATOR
1280           005346'                    .EVEN
```

```
                                        .SBTTL  SUBROUTINES FOR RP02/RP03 DEVICE ROUTINE
                                        ;SAVE REGISTERS R0 THRU R5
                                                ;JSR    R0,SAVREG       S/R CALL
005346' 010146                  SAVREG: MOV     R1,-(SP)                ;SAVE R0 THRU R5
005350' 010246                          MOV     R2,-(SP)
005352' 010346                          MOV     R3,-(SP)
005354' 010446                          MOV     R4,-(SP)
005356' 010546                          MOV     R5,-(SP)
005360' 010007                          MOV     R0,PC                   ;EXIT IN-LINE
                                        ;RESTORE REGISTERS R0 THRU R5
                                                ;JSR    R0,RESREG       S/R CALL
005362' 005726                  RESREG: TST     (SP)+                   ;RESTORE R5 THRU R0
005364' 012605                          MOV     (SP)+,R5
005366' 012604                          MOV     (SP)+,R4
005370' 012603                          MOV     (SP)+,R3
005372' 012602                          MOV     (SP)+,R2
005374' 012601                          MOV     (SP)+,R1
005376' 000200                          RTS     R0                      ;EXIT IN-LINE
                                        ;SET PROGRAM'S PROG TABLE ADR IN R3 & RPCS ADR IN R4
                                                ;JSR    PC,SUPTAD       S/R CALL
005400' 010703                  SUPTAD: MOV     PC,R3                   ;SET UP LOCATION ZERO ADR
005402' 062703  172376                  ADD     #LOCZ-.,R3
005406' 166303  177776                  SUB     -2(R3),R3               ;SUBTRACT PROG TBL LENGTH
005412' 016704  172406                  MOV     DREGAD,R4               ;GET DEV REG BASE ADR
005416' 062704  000004                  ADD     #4,R4                   ;POINT AT RPCS REGISTER
005422' 000207                          RTS     PC                      ;EXIT IN-LINE
                                        ;STORE DEVICE'S STATUS REGISTERS
                                                ;JSR    R5,STSTAT       S/R CALL
                                                ;.WORD  STADR-.         REL STORAGE ADR
                                                ;DESTROYS R0,R1
005424' 010501                  STSTAT: MOV     R5,R1                   ;GET REL STORAGE ADR & MAKE
005426' 062501                          ADD     (R5)+,R1                ;IT ABSOLUTE
005430' 016700  172370                  MOV     DREGAD,R0               ;GET DEV REG ADR
005434' 012702  000013                  MOV     #REGNUM,R2              ;SET UP # OF REGS
005440' 012021                  10$:    MOV     (R0)+,(R1)+             ;STORE DEV REG
005442' 005302                          DEC     R2                      ;DONE ALL REGS?
005444' 001375                          BNE     10$                     ;Y,N-10$
005446' 000205                          RTS     R5                      ;EXIT IN-LINE
```

```
1338
1339                                    ;DISPLAY DEVICE I.D. & UNIT #
1340
1341                                            ;JSR     PC,DEVID        S/R CALL
1342                                            ;
1343                                            ;R3 = PROG TBL ADR
1344                                            ;
1345                                            ;DESTROYS R0,R1,R2
1346
1347 005450' 012700 031060             DEVID:  MOV     #"02,R0         ;INITIALIZE TO RP02
1348 005454' 032763 000020 000032               BIT     #20,PHDLCD(R3)  ;IS THIS AN RP02?
1349 005462' 001402                             BEQ     10$             ;N,Y-10$
1350 005464' 012700 031460                      MOV     #"03,R0         ;SET UP FOR AN RP03
1351 005470' 010067 000516             10$:     MOV     R0,UNITMG+6     ;TAILOR DEV I.D. MSG
1352 005474' 012767 000026 000056               MOV     #22.,DEVIML     ;INITIALIZE TO NORMAL MSG LNGTH
1353 005502' 116300 000035                      MOVB    PCURDV(R3),R0   ;GET CURR UNIT #
1354 005506' 020027 000007                      CMP     R0,#7           ;VALID UNIT #?
1355 005512' 101007                             BHI     DEVIIV          ;Y,N-DEVIIV
1356 005514' 004577 172340                      JSR     R5,@#TASLZ      ;CONVERT # TO DECIMAL ASCII
1357 005520' 000510                             .WORD   UNASCI-.
1358 005522' 016767 000506 000500               MOV     UNASCI+4,UNASCI ;MOVE ASCII # TO 1ST TWO DIGITS
1359 005530' 000410                             BR      DEVIPR          ;GO ISSUE MSG
1360 005532' 012767 000032 000020 DEVIIV:       MOV     #26.,DEVIML     ;SET UP ERR COND MSG LNGTH
1361 005540' 042700 177400                      BIC     #177400,R0      ;RESET HIGH BYTE
1362 005544' 004577 172306                      JSR     R5,@#INASC      ;CONVERT BINARY # TO ASCII
1363 005550' 000460                             .WORD   UNASCI-.
1364 005552' 004567 000242             DEVIPR:  JSR     R5,PRINT        ;GO ISSUE UNIT # MSG
1365 005556' 000426                             .WORD   UNITMG-.
1366 005560' 000026                     DEVIML: .WORD   22.
1367 005562' 000207                             RTS     PC              ;EXIT IN-LINE
1368
1369
1370                                    ;TAILOR STATUS MSG & PRINT IT
1371
1372                                            ;JSR     R5,DISPST      S/R CALL
1373                                            ;WORD    STATADR-.      REL ADR OF STATUS DATA
1374                                            ;DESTROYS R0,R1,R2
1375
1376 005564' 010346                     DISPST: MOV     R3,-(SP)        ;SAVE R3
1377 005566' 010503                             MOV     R5,R3           ;GET REL DATA ADR
1378 005570' 062503                             ADD     (R5)+,R3        ;MAKE IT ABS
1379 005572' 010546                             MOV     R5,-(SP)        ;SAVE R5
1380 005574' 010705                             MOV     PC,R5           ;SET UP ADR OF REG NAMES IN ASCII
1381 005576' 062705 172320                      ADD     #DVREGS-.,R5
1382 005602' 012746 000013                      MOV     #11.,-(SP)      ;SET UP # OF REGISTERS TO DISPLAY
1383 005606' 012700 000003             10$:     MOV     #3,R0           ;SET UP 3 REG LOOP CNT
1384 005612' 010701                             MOV     PC,R1           ;POINT AT REG NAME IN MSG
1385 005614' 062701 000422                      ADD     #DVRGMG-.,R1
1386 005620' 012521                    15$:     MOV     (R5)+,(R1)+     ;MOVE REG NAME TO MSG
1387 005622' 012521                             MOV     (R5)+,(R1)+
1388 005624' 005725                             TST     (R5)+           ;POINT TO NEXT NAME
1389 005626' 062701 000012                      ADD     #10.,R1         ;POINT TO NEXT FIELD IN MSG
1390 005632' 005300                             DEC     R0              ;DONE 3 REGS?
1391 005634' 001371                             BNE     15$             ;Y,N-15$
1392 005636' 012300                             MOV     (R3)+,R0        ;CONVERT OCTAL REGISTER CONTENTS
1393 005640' 004577 172212                      JSR     R5,@#INASC      ;FOR 3 REGISTERS TO ASCII
```

```
1394  005644' 000400                          .WORD   DVRDT1-.                ;AND PLACE IN THE MSG
1395  005646' 012300                          MOV     (R3)+,R0
1396  005650' 004577 172202                   JSR     R5,@#INASC
1397  005654' 000406                          .WORD   DVRDT2-.
1398  005656' 012300                          MOV     (R3)+,R0
1399  005660' 004577 172172                   JSR     R5,@#INASC
1400  005664' 000414                          .WORD   DVRDT3-.
1401  005666' 012767 000050 000034            MOV     #40.,30$                ;INITIALIZE MSG LENGTH TO 3 REGS
1402  005674' 162716 000003                   SUB     #3,(SP)                 ;DECR REGISTER CNT
1403  005700' 100005                          BPL     25$                     ;< 3 REGS? (Y,N-25$)
1404  005702' 162767 000016 000020  20$:      SUB     #14.,30$                ;SHORTEN MSG LENGTH BY 1 REG
1405  005710' 005216                          INC     (SP)                    ;INCR NEG REG CNT
1406  005712' 100773                          BMI     20$                     ;CNT BACK TO 0? (Y,N-20$)
1407  005714' 010346                 25$:     MOV     R3,-(SP)                ;SAVE REG DATA PNTR
1408  005716' 016603 000006                   MOV     6(SP),R3                ;RESTORE PROG TBL ADR
1409  005722' 004567 000072                   JSR     R5,PRINT                ;GO PRINT THE MSG
1410  005726' 000310                          .WORD   DVRGMG-.
1411  005730' 000050                  30$:     .WORD   40.
1412  005732' 012603                          MOV     (SP)+,R3                ;RESTORE REG DATA PNTR
1413  005734' 005716                          TST     (SP)                    ;MORE REGS TO GO?
1414  005736' 001323                          BNE     10$                     ;N,Y-10$
1415  005740' 005726                          TST     (SP)+                   ;REMOVE CNT FROM STACK
1416  005742' 012605                          MOV     (SP)+,R5                ;RESTORE R5 & R3
1417  005744' 012603                          MOV     (SP)+,R3
1418  005746' 000205                          RTS     R5                      ;EXIT IN-LINE
1419
1420
1421                                  ;DISPLAY CYL/HEAD/SECT WORDS' VALUES
1422
1423                                          ;JSR    PC,PRTIWD       S/R CALL
1424                                          ;DESTROYS R0,R1,R2
1425
1426  005750' 016700 172030          PRTIWD:  MOV     CYL,R0                  ;GET CYL VALUE
1427  005754' 004577 172076                   JSR     R5,@#INASC              ;CONVERT ITS VALUE TO ASCII
1428  005760' 000773                          .WORD   IFCYL-.
1429  005762' 016700 172020                   MOV     HEAD,R0                 ;GET & CONVERT HEAD VALUE
1430  005766' 004577 172064                   JSR     R5,@#INASC
1431  005772' 000776                          .WORD   IFHEAD-.
1432  005774' 016700 172010                   MOV     SECT,R0                 ;GET & CONVERT SECT VALUE
1433  006000' 004577 172052                   JSR     R5,@#INASC
1434  006004' 001001                          .WORD   IFSECT-.
1435  006006' 004567 000006                   JSR     R5,PRINT                ;PRINT MSG WITH THEIR VALUES
1436  006012' 000734                          .WORD   INFOMG-.
1437  006014' 000045                          .WORD   37.
1438  006016' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                        ;ISSUE MSG TO LIST DEVICE SUBROUTINE
                                        ;
                                        ;JSR    R5,PRINT        S/R CALL
                                        ;.WORD  MSGADR-.        REL ADR OF MSG
                                        ;.WORD  BYTCNT          MSG BYTE CNT (IF NEGATIVE,
                                        ;                         RESET PRT DEV DEDICATED.)
                                        ;R3 = PROG TBL ADR
                                        ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
                                        ;DESTROYS R0,R1,R2
                                        ;
006020' 010500          PRINT:  MOV     R5,R0                   ;GET MSG ADR & MAKE IT ABS
006022' 062500                  ADD     (R5)+,R0
006024' 012501                  MOV     (R5)+,R1                ;GET BYTE COUNT
006026' 005704                  TST     R4                      ;USE CMND MODE PRINT?
006030' 100030                  BPL     40$                     ;Y,N-40$
006032' 010702                  MOV     PC,R2                   ;SET UP LINK INFO ADR
006034' 062702  000040          ADD     #20$-.,R2
006040' 160200                  SUB     R2,R0                   ;MAKE MSG ADR REL
006042' 010022                  MOV     R0,(R2)+                ;STORE MSG ADR
006044' 010112                  MOV     R1,(R2)                 ;STORE MSG'S BYTE COUNT
006046' 100001                  BPL     10$                     ;CNT NEG? (Y,N-10$)
006050' 005412                  NEG     (R2)                    ;MAKE IT POSITIVE
006052' 016367  000006  000056  10$:    MOV     PASCIN(R3),PROGNM       ;STORE PROG'S # IN MSG
006060' 004577  171770          JSR     R5,@CLIST               ;ISSUE PROG #
006064' 000050                  .WORD   PNMMSG-.
006066' 000005                  .WORD   5
006070' 004577  171760          JSR     R5,@CLIST               ;ISSUE MSG SPECIFIED
006074' 000000          20$:    .WORD   XXXX
006076' 000000                  .WORD   XXXX
006100' 004577  171750          JSR     R5,@CLIST               ;ISSUE A <CR> & <LF>
006104' 000254                  .WORD   CRLF-.
006106' 000002                  .WORD   2
006110' 000410                  BR      PRTEX                   ;GO TO EXIT
006112' 010067  000010  40$:    MOV     R0,50$                  ;STORE MSG'S ABS ADR
006116' 010167  000006          MOV     R1,60$                  ;STORE ITS BYTE CNT
006122' 004577  171724          JSR     R5,@ULIST               ;GO TO MPG TO ISSUE THE MSG
006126' 000000          50$:    .WORD   XXXX
006130' 000000          60$:    .WORD   XXXX
006132' 000205          PRTEX:  RTS     R5                      ;EXIT IN-LINE
```

```
                                                .SBTTL  RP02/RP03 MESSAGE STORAGE AREA
                                                .NLIST  BEX
                                                .EVEN
006134' 021520                          PNMMSG: .ASCII  /P#/
006136' 054130          011             PROGNM: .ASCII  /XX/<011>
006141' 000101  020124  040514          ATIMSG: .ASCII  /AT LAST INT:/
006155' 000103  051125  042522          CURMSG: .ASCII  /CURRENTLY:/
006167' 000105  042116  047440          RENDMG: .ASCII  /END OF REPORT/
                                                .EVEN
006204' 025052  025052  050122          UNITMG: .ASCII  /####RPXX DISK UNIT: /
006230' 054130  054130  054130          UNASCI: .ASCII  /XXXXXX/
                                                .EVEN
006236' 054130  054130  020075          DVRGMG: .ASCII  /XXXX= /
006244' 054130  054130  054130          DVRDT1: .ASCII  /XXXXXX  XXXX= /
006262' 054130  054130  054130          DVRDT2: .ASCII  /XXXXXX  XXXX= /
006310' 054130  054130  054130          DVRDT3: .ASCII  /XXXXXX/
006316' 054502  042524  035123          CNTSMG: .ASCII  /BYTES:  RD= /
006322' 054130  054130  054130          BCMRD:  .ASCII  /XXXXXXXXXXXX  WR= /
006344' 054130  054130  054130          BCMWR:  .ASCII  /XXXXXXXXXXXX/
006360' 005015                          CRLF:   .ASCII  <015><012>
006362' 044411  045503  020075                  .ASCII  <011><011>/CK= /
006370' 054130  054130  054130          BCMCK:  .ASCII  /XXXXXXXXXXXX/<015><012><011>/CMNDS:  RD= /
006423' 000130  054130  054130          CMDCRD: .ASCII  /XXXXXX WR= /
006436' 054130  054130  054130          CMDCWR: .ASCII  /XXXXXX CK= /
006451' 000130  054130  054130          CMDCCK: .ASCII  /XXXXXX/<015><012><011><011>/SK= /
006467' 000130  054130  054130          CMDCSK: .ASCII  /XXXXXX MISC= /
006504' 054130  054130  054130          CMDCMS: .ASCII  /XXXXXX/<015><012><011>/ERRORS: DEV= /
006532' 054130  054130  054130          CNTERR: .ASCII  /XXXXXX DATA= /
006547' 000130  054130  054130          CNTDER: .ASCII  /XXXXXX/<015><012><011>/RETRYS: TIMEE= /
006577' 000130  054130  054130          CNTTIE: .ASCII  /XXXXXX CSME= /
006614' 054130  054130  054130          CNTCSE: .ASCII  /XXXXXX WPE= /
006630' 054130  054130  054130          CNTWPE: .ASCII  /XXXXXX/<015><012><011><011>/LPE= /
006647' 000130  054130  054130          CNTLPE: .ASCII  /XXXXXX WCE= /
006663' 000130  054130  054130          CNTWCE: .ASCII  /XXXXXX/<015><012><011>/TOTAL RETRYS: /
006712' 054130  054130  054130          CNTRTY: .ASCII  /XXXXXX/<015><012><011>/INTERRUPTS:  /
006740' 054130  054130  054130          CNTINT: .ASCII  /XXXXXX/
                006746'                 CNTSEN= .
006746' 054503  036514  000040          INFOMG: .ASCII  /CYL= /
006753' 000130  054130  054130          IFCYL:  .ASCII  /XXXXXX HEAD= /
006770' 054130  054130  054130          IFHEAD: .ASCII  /XXXXXX SECT= /
007005' 000130  054130  054130          IFSECT: .ASCII  /XXXXXX/
                007014'                         .EVEN
007014' 052123  047115  020124          STMNMG: .ASCII  /STMNT # /
007024' 054130  054130  054130          STMNUM: .ASCII  /XXXXXX/
007032' 051105  047522  020122          ERRMSG: .ASCII  /ERROR BITS:/<015><012><011>
007050' 000100                          COOFLD: .BLKB   64.
007150' 027524  020117  047117          CRTO:   .ASCIZ  'T/O ON IDLE/CRESET'
007173' 000124  046511  047505          IOTO:   .ASCIZ  'TIMEOUT ON I/O'
007212' 047125  040523  042506          INITUS: .ASCII  'UNSAFE ON INITIATION'
007236' 027511  020117  042524          IOTERM: .ASCIZ  'I/O TERMINATION ERROR'
007264' 052101  047124  047040          NOATA:  .ASCIZ  'ATTN NOT SET'
                                                .ODD
007301' 000105  044130  052501          RTYEXH: .ASCII  'EXHAUSTED RETRIES ON '
```

```
 1536  007326' 054130  054130     000  RTYID:  .ASCIZ  'XXXX'
 1537  007333'    125  044516  020124  UNOFFL: .ASCIZ  'UNIT OFF-LINE'
 1538  007351'    125  044516  020124  UNNRDY: .ASCIZ  'UNIT NOT RDY'
 1539  007366' 047111  020126  047125  INVDVN: .ASCIZ  /INV UNIT #/
 1540          007402'                         .EVEN
 1541                                          .LIST   BEX
 1542
 1543          007402'                 DVREND= .
```

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

                        ;       PROGRAM TABLE FORMAT

000242                  PTLGTH= 162.    ;PROGRAM TABLE LENGTH - NON MEM MGMNT VERSION OF MPG

                        ;(PTLGTH= 212.  ;PROGRAM TABLE LENGTH - MEM MGMNT VERSION OF MPG)

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

000002                          URSTOP= 2       ; 1 = USER HAS STOPPED THIS PROGRAM
000004                          ERSTOP= 4       ; 1 = AN ERROR HAS STOPPED THIS PROGRAM
000010                          WT4IOT= 10      ; 1 = WAITING FOR I/O TERMINATION
000020                          CTPRIO= 20      ; 1 = CONSOLE OR PRINTER I/O IN PROGRESS
000040                          SETDED= 40      ; 1 = THIS PROG SET THE PRT DEV DEDICATED FLAG
000100                          OCPRES= 100     ; 1 = OBJ CODE IS PRESENT
000200                          USEUBM= 200     ; 1 = THIS PROG USES THE UNIBUS MAP (MEM MGMNT ONLY)
100000                          ACTIVE= 100000  ; 1 = PROGRAM IS ACTIVE (SPECIFIED FOR EXECUTION)

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

100000                          STONER= 100000  ; 1 = STOP PROG EXECUTION UPON ERROR
040000                          CYCPRG= 40000   ; 1 = CYCLE PROGRAM (ON CURRENT DEVICE)
020000                          PRONER= 20000   ; 1 = DO NOT PRINT ON ERROR
010000                          BIT12=  10000   ; 0 = NOT USED
004000                          BIT11=  4000    ; 0 = NOT USED
002000                          CYCDVL= 2000    ; 1 = CYCLE THE DEVICE LIST
001000                          GTNXTD= 1000    ; 1 = CYCLE ON SAME DEVICE UPON ERROR
000400                          DOERCK= 400     ; 1 = DON'T DO ERROR CHECKING
000200                          SPOPER= 200     ; 1 = DEVICE SPECIAL OPERATION
000100                          BIT6=   100     ; 0 = NOT USED
000040                          DOIOT=  40      ; 1 = DO NOT PERFORM I/O TIMEOUT
000020                          AUTORP= 20      ; 1 = DO NOT AUTOMATICALLY DISPLAY COUNTS
000010                          AURPEP= 10      ; 1 = AUTO DISPLAY COUNTS AT END OF FINAL PASS ONLY
000004                          HSKPEP= 4       ; 1 = HOUSEKEEP COUNTS ONLY AT RUN COMMAND
000002                          PFBBOV= 2       ; 1 = PRINT FIRST BAD BYTE ONLY ON VERIFY
000001                          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

000026                  PLNGTH= +22.    ;PROG AREA LENGTH (OBJ END MINUS PROG TBL START) - 1 WORD

000030                  PTOCNT= +24.    ;I/O TIMEOUT COUNT - 1 WORD
```

```
1601            000032                  PMDLCD= +26.    ;DEV ROUT MODEL # CODE - 1 WORD
1602
1603            000034                  PDPNTR= +28.    ;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1604
1605            000035                  PCURDV= +29.    ;CURRENT DEVICE # - 1 BYTE
1606
1607            000036                  PDNUMS= +30.    ;DEVICE NUMBERS - 16 BYTES
1608
1609            000056                  PTEM0=  +46.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1610
1611            000060                  PTEM1=  +48.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1612
1613            000062                  PTEM2=  +50.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1614
1615            000064                  PTEM3=  +52.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1616
1617            000066                  PTEM4=  +54.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1618
1619            000070                  PTEM5=  +56.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1620
1621            000072                  PTEM6=  +58.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1622
1623            000074                  PTEM7=  +60.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1624
1625            000076                  PTEM8=  +62.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1626
1627            000100                  PTEM9=  +64.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1628
1629            000102                  PTEM10= +66.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1630
1631            000104                  PTEM11= +68.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1632
1633            000106                  PTEM12= +70.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1634
1635            000110                  PTEM13= +72.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1636
1637            000112                  PTEM14= +74.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1638
1639            000114                  PTEM15= +76.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1640
1641            000116                  PNBR=   +78.    ;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1642
1643            000120                  PSRC=   +80.    ;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1644
1645            000122                  PDST=   +82.    ;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1646
1647            000124                  PSTKCT= +84.    ;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1648
1649            000126                  PSTKSV= +86.    ;STACK WORDS STORAGE AREA - 30 WORDS
1650
1651            000222                  PSVREG= +146.   ;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1652
1653            000236                  PUSRPC= +158.   ;USER'S CURRENT PROGRAM COUNTER - 1 WORD
1654
```

```
1656                                    ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
1657
1658                                ;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
1659
1660                                ;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
1661
1662                                ;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
1663
1664                                ;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
1665
1666                                ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
1667
1668                                ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
1669
1670                                ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
1671
1672                                        ;END OF MEM MGMNT ONLY ENTRIES
1673
1674             000240             PTSIZE= +160.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
1675
1676                                ;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
1677
1678             000242             PTEND=  +162.   ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
1679
1680                                ;(PTEND=  +212. ;END OF PROGRAM TABLE - MEM MGMNT VERSION)
```

```
                        ;       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
```

```
1738            000062                  DVPDTA= +50.    ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1739
1740            000064                  DVSFWD= +52.    ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1741
1742            000066                  DVSVEC= +54.    ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1743
1744            000070                  DVCVEC= +56.    ;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1745
1746            000072                  DVTVEC= +58.    ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1747
1748            000074                  DVRINT= +60.    ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1749
1750            000076                  DVGETB= +62.    ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1751
1752            000100                  DVPUTB= +64.    ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1753
1754            000102                  DEVSTP= +66.    ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1755
1756            000104                  DEVETP= +68.    ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1757
1758            000106                  DVPTEP= +70.    ;PACK TABLE EXTEN. REL POINTER - 1 WORD
1759
1760            000110                  DVVTEP= +72.    ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1761
1762            000112                  DVCTEP= +74.    ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
1763
1764            000114                  DVIWSP= +76.    ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1765
1766            000116                  DRTEND= +78.    ;END OF DEVICE ROUTINE TABLE
1767
1768
1769            000001                          .END
```

```
ACTIVE= 100000
ATATBL  003216R  002
ATINTSG 006141R  002
ATNINT  003476R  002
AURPEP= 000010
AUTORP= 000020
BCMCK   006370R  002
BCMRD   006322R  002
BCMWR   006344R  002
BINASC  000056R  002
BIT11 = 004000
BIT12 = 010000
BIT6  = 000100
BTASLZ  000060R  002
BYCK    001276R  002
BYRD    001266R  002
BYWR    001272R  002
CIOBSY  000046R  002
CKCNT   001306R  002
CKOBSY  004076R  002
CKRTRY  003636R  002
CLIST   000054R  002
CLRVEC  000070R  002
CLRWTF  003462R  002
CMDCCK  006451R  002
CMDCMS  006504R  002
CMDCOM  002646R  002
CMDCRD  006423R  002
CMDCSK  006467R  002
CMDCWR  006436R  002
CNTADR  001244R  002
CNTCSE  006614R  002
CNTDER  006547R  002
CNTERR  006532R  002
CNTINT  006740R  002
CNTLPE  006647R  002
CNTNUM= 000024
CNTRTY  006712R  002
CNTSEN= 006746R  002
CNTSMG  006306R  002
CNTTIE  006577R  002
CNTWCE  006663R  002
CNTWPE  006630R  002
CODFLD  007050R  002
COUNTS  001266R  002
CRESET  002402R  002
CRLF    006360R  002
CRTO    007150R  002
CSECNT  001322R  002
CSTAT   001214R  002
CSYSFW  000064R  002
CTPRIO= 000020
CUPGER  000050R  002
CURADR  001252R  002
CURCMD  001250R  002
CURCNT  001256R  002
CURFLG  001246R  002
CURMSG  006455R  002
CURRTY  001262R  002
CYCDVL= 002000
CYCPRG= 040000
CYL     000004R  002
DATAER  001316R  002
DECRSC  000062R  002
DECTAD= 000042
DERPAD= 000036
DEVBTA= 000056
DEVDER= 000050
DEVDRA= 000024
DEVETP= 000104
DEVFWD= 000002
DEVID   005450R  002
DEVIIV  005532R  002
DEVIML  005560R  002
DEVIOB= 000046
DEVIPR  005552R  002
DEVIVA= 000026
DEVIW1= 000004
DEVIW2= 000006
DEVIW3= 000010
DEVIW4= 000012
DEVIW5= 000014
DEVIW6= 000016
DEVIW7= 000020
DEVIWB= 000022
DEVRPS= 000130
DEVRSZ= 000000
DEVSTP= 000102
DEVWPS= 000032
DFLGWD  000002R  002
DHKPAD= 000034
DISCNT  001614R  002
DISPST  005564R  002
DKILAD= 000040
DOERCK= 000400
DOIOT = 000040
DOTERM= 000002
DREGAD  000024R  002
DRTEND= 000116
DRTLTH= 000116
DTOEAD= 000044
DVBTDA= 000060
DVCMDS  000226R  002
DVCPRT= 000054
DVCPTE  000700R  002
DVCTEP= 000112
DVCVEC= 000070
DVGETB= 000076
DVIWSP= 000114
DVIWST  001106R  002
DVMVTE  000570R  002
DVPDTA= 000062
DVPKTE  000350R  002
DVPTEP= 000106
DVPUTB= 000100
DVRDT1  006244R  002
DVRDT2  006262R  002
DVRDT3  006300R  002
DVREGE= 000226R  002
DVREGS  000116R  002
DVREMD= 007402R  002
DVREX   001714R  002
DVRGMG  006236R  002
DVRINT= 000074
DVSFWD= 000064
DVSVEC= 000066
DVTVEC= 000072
DVUPRT= 000052
DVVTEP= 000110
EBSBAS  004564R  002
EBSTAT  004566R  002
ERRMSG  007032R  002
ERRDTB  005036R  002
ERRBAS  004532R  002
ERR   = 100000
ERRCNT  001314R  002
ERRCOM  004440R  002
ERRCS   004400R  002
ERRCS1  004406R  002
ERREX   005034R  002
ERRFND  003564R  002
ERRI    000022R  002
ERRIS   004424R  002
ERRSNM  004754R  002
ERSTAD  004746R  002
ERSTOP= 000004
FINCNT  001260R  002
GETBYT  000076R  002
GTNXTD= 001000
HDROFF  002352R  002
HDRON   002342R  002
HE    = 040000
HEAD    000006R  002
HEADER= 004000
HOMESK  002640R  002
HSKEEP  001364R  002
HSKPEN= 001336R  002
HSKPEP= 000004
HSKPST= 001166R  002
IDLE    002402R  002
IFCYL   006753R  002
IFHEAD  006770R  002
IFSECT  007005R  002
INFOMG  006746R  002
INITUS  007212R  002
INTCNT  001334R  002
INTEAD  004272R  002
INTEX   003466R  002
INVOVN  007366R  002
IOERR = 000001
IOTERM  007236R  002
IOTO    007173R  002
ISTAT = 001166R  002
IVCTRD  000026R  002
KILL    002054R  002
KILLEX  002110R  002
LOCZ    000000R  002
LPECNT  001326R  002
MISCNT  001312R  002
MMVER = 000001
MODE  = 010000
MODE10  002372R  002
MODE11  002362R  002
MSFMT1  001140R  002
MSFMT2  001141R  002
MSFMT3  001146R  002
MSFMT4  001155R  002
MSFMT5  001164R  002
MYATA   003236R  002
NOATA   007264R  002
NOCOMP= 000001
NOWAIT  002332R  002
OBJAOR  001242R  002
OCPRES= 000100
OTHATA  003240R  002
PASCIN= 000006
PATCH   001340R  002
PC    =%000007
PCUROV= 000035
PDNUMS= 000036
PDPNTR= 000034
POST  = 000122
PFBBOV= 000002
PFLGWD= 000000
PFWADR= 000004
PLNGTH= 000026
PMDLCD= 000032
PNAME = 000010
PNBR  = 000116
PNMMSG  006134R  002
POBJST= 000024
POPSW = 000002
PRDIOA= 000016
PRINT   006020R  002
PROCEX  004312R  002
PROCTM  004176R  002
PROGNM  006136R  002
PRONER= 020000
PRTEX   006132R  002
PRTIWD  005750R  002
PSRC  = 000120
PSRCST= 000022
PSTKCT= 000124
PSTKSV= 000126
PSVREG= 000222
PSWD    000030R  002
PTEM0 = 000056
PTEM1 = 000060
PTEM10= 000102
PTEM11= 000104
PTEM12= 000106
PTEM13= 000110
PTEM14= 000112
PTEM15= 000114
PTEM2 = 000062
PTEM3 = 000064
PTEM4 = 000066
PTEM5 = 000070
PTEM6 = 000072
PTEM7 = 000074
PTEM8 = 000076
PTEM9 = 000100
PTEN0 = 000242
PTLGTH= 000242
PTOCNT= 000030
PTSIZE= 000240
PUSRPC= 000236
PUTBYT  000100R  002
PWRIOA= 000020
RDCNT   001302R  002
RDCOM   002472R  002
RDNOSK  002542R  002
READ    002462R  002
RECAL   002640R  002
REGNUM= 000013
RENDMG  006167R  002
REPORT  001436R  002
REPTBL  001724R  002
RESREG  005362R  002
RETRYS  001332R  002
RINTEX  004346R  002
RINTV   004324R  002
RPBA  = 000004
RPCA  = 000006
RPCS  = 000000
```

DTR3AA.P11 SYMBOL TABLE

```
RPCSV   001336R  002    RTYEXH  007301R  002    SIZE    000020R  002    SUCA  = 000020          WAIT    002312R  002
RPDA  = 000010          RTYID   007326R  002    SKCNT   001310R  002    SUIORG  003242R  002    WAITMO= 100000
RPDS  = 177774          R0    =X000000         SKCOM   002622R  002    SUPTAD  005400R  002    WCECNT  001330R  002
RPER  = 177776          R1    =X000001         SP    =X000006           TIECNT  001320R  002    WPECNT  001324R  002
RPINT   003354R  002    R2    =X000002         SPOPER= 000200             TOUTER  001774R  002    WRCK    002566R  002
RPM1  = 000012          R3    =X000003         STEPDN  002244R  002    TSTVEC  000072R  002    WRCNT   001304R  002
RPM2  = 000014          R4    =X000004         STEPEX  002160R  002    TVECT   004350R  002    WRCOM   002522R  002
RPM3  = 000016          R5    =X000005         STEPUP  002112R  002    TVECTX  004376R  002    WRITE   002512R  002
RPTBAS  001660R  002    SAVREG  005346R  002    STMADR  004162R  002    ULIST   000052R  002    WRNOSK  002554R  002
RPTEND  001704R  002    SECT    000010R  002    STMNMG  007014R  002    UNASCI  006230R  002    WT4IOT= 000010
RPTLP   001642R  002    SEEK    002616R  002    STMNUM  007024R  002    UNITMG  006204R  002    WWATTN= 000004
RPWC  = 000002          SETDED= 000040           STONER= 100000             UNNRDY  007351R  002    XXXX  = 000000
RTNINT  000074R  002    SETERR  003454R  002    STPCOM  002176R  002    UNOFFL  007333R  002    .     = 007402R  002

. ABS.  000000  000
        000000  001
RP11    007402  002

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 *,DTR3AA/NL:TOC/DOC=DTR3AA.P11
 RUN-TIME: 5 11 1 SECONDS
 RUN-TIME RATIO: 21/17=1.2
 CORE USED:  5K  (9 PAGES)

 DOCUMENT PAGES:  41
```

D04