DV11

DEVICE ROUTINE (MPG)

MD-11-DTDUA-A

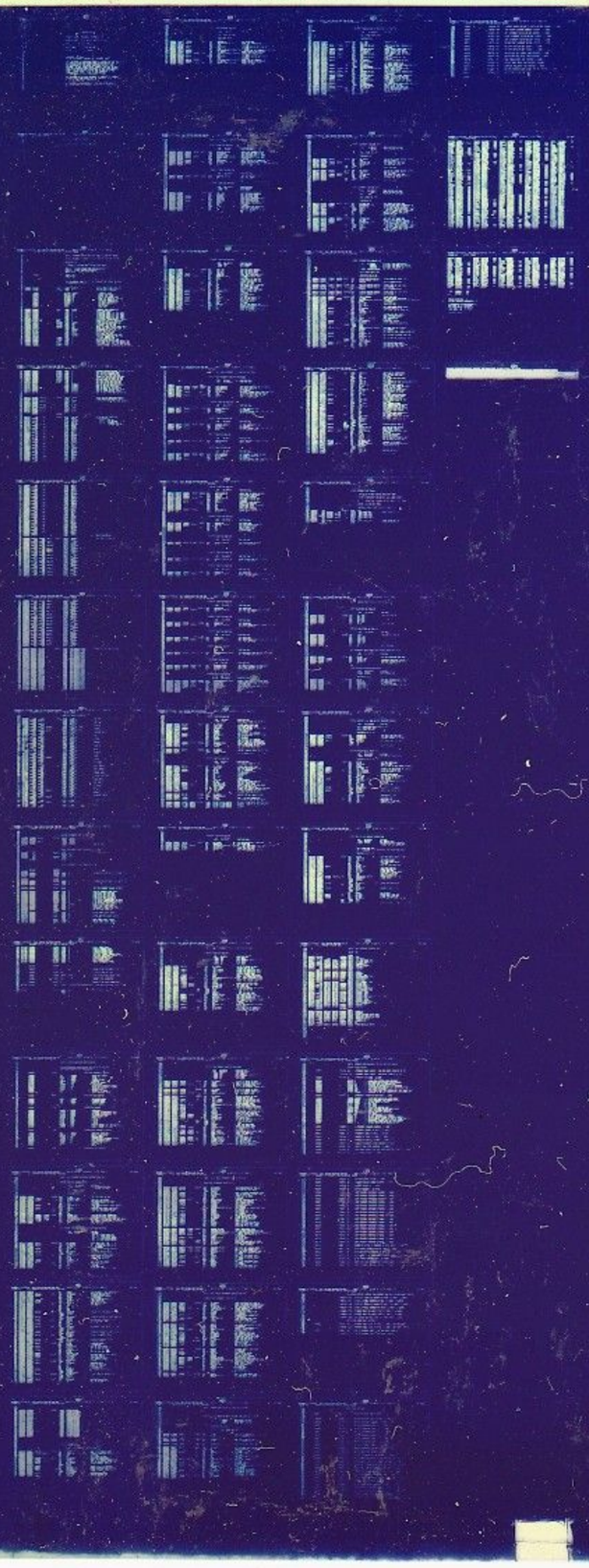
EP-DTDUA-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN U.S.A.



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DTDUA-A

PRODUCT NAME: DU11 DEVICE ROUTINE FOR MPG

DATE: JULY 1976

MAINTAINER: SYSTEMS RELIABILITY

AUTHOR: C. E. HARPER

%

```
                                                .SBTTL  REVISION HISTORY

                                        ;       JUL 76  DTDUA-A INITIAL RELEASE
```

```
                                        .SBTTL  STANDARD DEVICE ROUTINE TABLE & CONSTANT AREAS

                                        .TITLE  MAINDEC-11-DTDUA-A  DU11 DEVICE ROUTINE FOR MPG

                                        ;REVISION "A"

                                        ;FILENAME OF "TDUAA0.MPG" ON MPG/XXDP MEDIA

                                        ;MACY11:  DTDUA?,DTDUA?/CRF:SYM/DOC=DTDUA?.P11
                                        ;LNKX11:  DTDUA?.MPG/B:0←DTDUA?/E
                                        ;PAPER TAPE:  PUNCH DTDUA?.MPG/FILE:ELEV

        000000'                         .CSECT  DU11
                                        .DSABL  GBL


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

000000' 007302          LOCZ:   .WORD   DVREND-.                ;DEVICE ROUT SIZE IN BYTES
000002' 000000          DFLGWD: .WORD   0                       ;DEVICE ROUT FLAGWORD

        100000                  WAITMD= 100000                  ;  'NOWAIT' FLAG
        000200                  BRKFLG= 200                     ;  BREAK INST FLAG
        000100                  WRIERR= 100                     ;  WRITE INT ERROR FLAG
        000040                  RDIDSC= 40                      ;  READ INT DATA SET CHG ERR
        000020                  RDIERR= 20                      ;  READ INT ERROR FLAG
        000010                  WRTERM= 10                      ;  DO WRITE TERMINATION
        000004                  RDTERM= 4                       ;  DO READ TERMINATION
        000002                  WRBSY=  2                       ;  WRITE BUSY
        000001                  RDBSY=  1                       ;  READ BUSY

000004' 000026          SYNC:   .WORD   SYNCB                   ;CURRENT SYNC CHARACTER
000006' 000005          SCNT:   .WORD   NSYNC                   ;SYNC CHAR COUNT
000010' 000000                  .WORD   0                       ;INTERFACE WORD # 3 (NOT USED)
000012' 000000                  .WORD   0                       ;INTERFACE WORD # 4 (NOT USED)
000014' 000000                  .WORD   0                       ;INTERFACE WORD # 5 (NOT USED)
000016' 000000                  .WORD   0                       ;INTERFACE WORD # 6 (NOT USED)
000020' 000000          SIZE:   .WORD   0                       ;# OF BYTES TRANSFERRED / UNIMAP FLG
000022' 000000          ERRI:   .WORD   0                       ;ERROR ON LAST I/O INDICATOR
000024' 160010          DREGAD: .WORD   160010                  ;FIRST DEVICE REGISTER ADR
000026' 000300          IVCTAD: .WORD   300                     ;INTERRUPT VECTOR ADR
000030' 000240          RDPSWD: .WORD   240                     ;READ INT PROC STATUS WORD (BR 5)
000032' 000240          WRPSWD: .WORD   240                     ;WRITE INT PROC STATUS WORD (BR 5)
000034' 001464                  .WORD   HSKEEP-.                ;HOUSEKEEPING ROUT REL ADR
000036' 001556                  .WORD   REPORT-.                ;REPORT ROUT REL ADR
000040' 002222                  .WORD   KILL-.                  ;KILL ROUT REL ADR
000042' 001376                  .WORD   DATAER-.                ;DATA ERROR COUNTER REL ADR
000044' 002126                  .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
```

```
                                DECASC: .WORD   0                       ;CONVERT PACKED DECIMAL TO ASCII BR ADR
                                CSYSFW: .WORD   0                       ;MPG SYSTEM FLAGWORD ADR
                                SETVEC: .WORD   0                       ;SET INT VECT ROUT BR ADR
                                CLRVEC: .WORD   0                       ;CLEAR INT VECTOR ROUT BR ADR
                                TSTVEC: .WORD   0                       ;TEST INT VECTOR ROUT BR ADR
                                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

                                DVREGS: .ASCII  /RCSR/                  ;VALID DEVICE REGISTER NAMES &
                                        .WORD   0                       ;THEIR POSITIONS RELATIVE TO
                                        .ASCII  /RBUF/                  ;THE DEVICE REGISTERS BASE ADDRESS.
                                        .WORD   2
                                        .ASCII  /TCSR/
                                        .WORD   4
                                        .ASCII  /TBUF/
                                        .WORD   6
                                        .ASCII  /PCSR/
                                        .WORD   2
                                DVREGE= .

                                DVCMDS: .BYTE   120,001                 ;VALID DEVICE FUNCTIONS
                                        .WORD   READ-.                  ;  FLAG BYTE:
                                        .BYTE   130,001                 ;    BIT 7 = NPR DEV
                                        .WORD   WRITE-.                 ;    BIT 3 = MASSBUS DEV
                                        .BYTE   160,001                 ;    BIT 0 = 2 WORDS FOR ADR
                                        .WORD   BREAK-.                 ;            (18 BIT ADRS)
                                        .BYTE   376,0
                                        .WORD   NOWAIT-.
                                        .BYTE   375,0
                                        .WORD   WAIT-.
                                        .BYTE   374,0
                                        .WORD   REPORT-.
                                        .BYTE   373,0
                                        .WORD   REPORT-.
                                        .BYTE   372,0
                                        .WORD   CRESET-.
                                        .BYTE   371,0
                                        .WORD   CALL-.
                                        .BYTE   370,0
                                        .WORD   LISTEN-.
                                        .BYTE   367,0
                                        .WORD   ANSWER-.
                                        .BYTE   366,0
                                        .WORD   HANGUP-.
                                        .BYTE   365,0
                                        .WORD   SEND-.
                                        .BYTE   364,0
                                        .WORD   RECV-.
```

```
000244'    363     000                         .BYTE   363,0
000246'  002734                                .WORD   MODE-.
000250'    362     000                         .BYTE   362,0
000252'  002542                                .WORD   STRIP-.
000254'    361     000                         .BYTE   361,0
000256'  002546                                .WORD   NSTRIP-.
000260'    360     000                         .BYTE   360,0
000262'  002552                                .WORD   FDUPLX-.
000264'    357     000                         .BYTE   357,0
000266'  002562                                .WORD   HDUPLX-.
000270'    356     000                         .BYTE   356,0
000272'  002572                                .WORD   NORMAL-.
000274'    355     000                         .BYTE   355,0
000276'  002602                                .WORD   SYSTST-.
000300'    354     000                         .BYTE   354,0
000302'  002612                                .WORD   EVEN-.
000304'    353     000                         .BYTE   353,0
000306'  002616                                .WORD   ODD-.
000310'    352     000                         .BYTE   352,0
000312'  002630                                .WORD   NOPAR-.
000314'    351     000                         .BYTE   351,0
000316'  002750                                .WORD   BITS-.
000320'    350     000                         .BYTE   350,0
000322'  003074                                .WORD   PRESET-.
000324'    347     000                         .BYTE   347,0
000326'  002140                                .WORD   GENPAR-.
000330'    346     000                         .BYTE   346,0
000332'  002074                                .WORD   CVSYNC-.
000334'    345     000                         .BYTE   345,0
000336'  002306                                .WORD   READY-.
000340'  177777                                .WORD   177777                  ;TABLE TERMINATOR

000342'  047516  040527  052111  DVPKTE:  .ASCII  /NOWAIT/                ;PACK TABLE EXTENSION
000350'    376     000                         .BYTE   376,0
000352'  020040  040527  052111           .ASCII  /  WAIT/
000360'    375     000                         .BYTE   375,0
000362'  052123  052101  051525           .ASCII  /STATUS/
000370'    374     000                         .BYTE   374,0
000372'  047503  047125  051524           .ASCII  /COUNTS/
000400'    373     000                         .BYTE   373,0
000402'  051103  051505  052105           .ASCII  /CRESET/
000410'    372     000                         .BYTE   372,0
000412'  020040  040503  046114           .ASCII  /  CALL/
000420'    371     000                         .BYTE   371,0
000422'  044514  052123  047105           .ASCII  /LISTEN/
000430'    370     000                         .BYTE   370,0
000432'  047101  053523  051105           .ASCII  /ANSWER/
000440'    367     000                         .BYTE   367,0
000442'  040510  043516  050125           .ASCII  /HANGUP/
000450'    366     000                         .BYTE   366,0
000452'  020040  042523  042116           .ASCII  /  SEND/
000460'    365     000                         .BYTE   365,0
000462'  020040  042522  053103           .ASCII  /  RECV/
000470'    364     000                         .BYTE   364,0
000472'  020040  047515  042504           .ASCII  /  MODE/
000500'    363     000                         .BYTE   363,0
```

```
000502' 051440 051124 050111          .ASCII  / STRIP/
000510'    362    000                 .BYTE   362,0
000512' 051516 051124 050111          .ASCII  /NSTRIP/
000520'    361    000                 .BYTE   361,0
000522' 042106 050125 054114          .ASCII  /FDUPLX/
000530'    360    000                 .BYTE   360,0
000532' 042110 050125 054114          .ASCII  /HDUPLX/
000540'    357    000                 .BYTE   357,0
000542' 047516 046522 046101          .ASCII  /NORMAL/
000550'    356    000                 .BYTE   356,0
000552' 054523 052123 052123          .ASCII  /SYSTST/
000560'    355    000                 .BYTE   355,0
000562' 020040 053105 047105          .ASCII  /  EVEN/
000570'    354    000                 .BYTE   354,0
000572' 020040 047440 042104          .ASCII  /   ODD/
000600'    353    000                 .BYTE   353,0
000602' 047040 050117 051101          .ASCII  / NOPAR/
000610'    352    000                 .BYTE   352,0
000612' 020040 044502 051524          .ASCII  /  BITS/
000620'    351    000                 .BYTE   351,0
000622' 051120 051505 052105          .ASCII  /PRESET/
000630'    350    000                 .BYTE   350,0
000632' 042507 050116 051101          .ASCII  /GENPAR/
000640'    347    000                 .BYTE   347,0
000642' 053103 054523 041516          .ASCII  /CVSYNC/
000650'    346    000                 .BYTE   346,0
000652' 051040 040505 054504          .ASCII  / READY/
000660'    345    000                 .BYTE   345,0

000662' 000376 001330  DVMVTE: .WORD   376,MSFMT1-LOCZ      ;MODEL VECTOR TABLE EXTEN.
000666' 000375 001330          .WORD   375,MSFMT1-LOCZ
000672' 000374 001330          .WORD   374,MSFMT1-LOCZ
000676' 000373 001330          .WORD   373,MSFMT1-LOCZ
000702' 000372 001330          .WORD   372,MSFMT1-LOCZ
000706' 000371 001330          .WORD   371,MSFMT1-LOCZ
000712' 000370 001330          .WORD   370,MSFMT1-LOCZ
000716' 000367 001330          .WORD   367,MSFMT1-LOCZ
000722' 000366 001330          .WORD   366,MSFMT1-LOCZ
000726' 000365 001330          .WORD   365,MSFMT1-LOCZ
000732' 000364 001330          .WORD   364,MSFMT1-LOCZ
000736' 000363 001327          .WORD   363,MSFMT2-LOCZ
000742' 000362 001330          .WORD   362,MSFMT1-LOCZ
000746' 000361 001330          .WORD   361,MSFMT1-LOCZ
000752' 000360 001330          .WORD   360,MSFMT1-LOCZ
000756' 000357 001330          .WORD   357,MSFMT1-LOCZ
000762' 000356 001330          .WORD   356,MSFMT1-LOCZ
000766' 000355 001330          .WORD   355,MSFMT1-LOCZ
000772' 000354 001330          .WORD   354,MSFMT1-LOCZ
000776' 000353 001330          .WORD   353,MSFMT1-LOCZ
001002' 000352 001330          .WORD   352,MSFMT1-LOCZ
001006' 000351 001327          .WORD   351,MSFMT2-LOCZ
001012' 000350 001330          .WORD   350,MSFMT1-LOCZ
001016' 000347 001324          .WORD   347,MSFMT3-LOCZ
001022' 000346 001324          .WORD   346,MSFMT3-LOCZ
001026' 000345 001330          .WORD   345,MSFMT1-LOCZ
```

```
                                        ;       COMPILER TABLE EXTENSION
                                        ;
001032' 003 376                 DVCPTE: .BYTE   3,376                   ;NO WAIT
001034' 004537 000012                   .WORD   4537,10.
001040' 003 375                         .BYTE   3,375                   ;WAIT
001042' 004537 000012                   .WORD   4537,10.
001046' 004 374                         .BYTE   4,374                   ;STATUS
001050' 004537 000012 001002            .WORD   4537,10.,1002
001056' 004 373                         .BYTE   4,373                   ;COUNTS
001060' 004537 000012 001001            .WORD   4537,10.,1001
001066' 003 372                         .BYTE   3,372                   ;CONTROL RESET
001070' 004537 000012                   .WORD   4537,10.
001074' 003 371                         .BYTE   3,371                   ;CALL
001076' 004537 000012                   .WORD   4537,10.
001102' 003 370                         .BYTE   3,370                   ;LISTEN
001104' 004537 000012                   .WORD   4537,10.
001110' 003 367                         .BYTE   3,367                   ;ANSWER
001112' 004537 000012                   .WORD   4537,10.
001116' 003 366                         .BYTE   3,366                   ;HANG UP
001120' 004537 000012                   .WORD   4537,10.
001124' 003 365                         .BYTE   3,365                   ;SEND
001126' 004537 000012                   .WORD   4537,10.
001132' 003 364                         .BYTE   3,364                   ;RECEIVE
001134' 004537 000012                   .WORD   4537,10.
001140' 004 363                         .BYTE   4,363                   ;MODE V
001142' 004537 000012 000000            .WORD   4537,10.,0
001150' 003 362                         .BYTE   3,362                   ;STRIP
001152' 004537 000012                   .WORD   4537,10.
001156' 003 361                         .BYTE   3,361                   ;NO STRIP
001160' 004537 000012                   .WORD   4537,10.
001164' 003 360                         .BYTE   3,360                   ;FULL DUPLEX
001166' 004537 000012                   .WORD   4537,10.
001172' 003 357                         .BYTE   3,357                   ;HALF DUPLEX
001174' 004537 000012                   .WORD   4537,10.
001200' 003 356                         .BYTE   3,356                   ;NORMAL MODE
001202' 004537 000012                   .WORD   4537,10.
001206' 003 355                         .BYTE   3,355                   ;SYSTEM TEST MODE
001210' 004537 000012                   .WORD   4537,10.
001214' 003 354                         .BYTE   3,354                   ;EVEN PARITY
001216' 004537 000012                   .WORD   4537,10.
001222' 003 353                         .BYTE   3,353                   ;ODD PARITY
001224' 004537 000012                   .WORD   4537,10.
001230' 003 352                         .BYTE   3,352                   ;NO PARITY
001232' 004537 000012                   .WORD   4537,10.
001236' 004 351                         .BYTE   4,351                   ;BITS V
001240' 004537 000012 000000            .WORD   4537,10.,0
001246' 003 350                         .BYTE   3,350                   ;PRESET
001250' 004537 000012                   .WORD   4537,10.
001254' 005 347                         .BYTE   5,347                   ;GENERATE PARITY V AT V
001256' 004537 000012 000000            .WORD   4537,10.,0,2
001264' 000002
001266' 005 346                         .BYTE   5,346                   ;CONVERT SYNC V AT V
001270' 004537 000012 000000            .WORD   4537,10.,0,2
001276' 000002
001300' 003 345                         .BYTE   3,345                   ;READY
001302' 004537 000012                   .WORD   4537,10.
```

```
                                        ;
                                        ;       DEVICE INTERFACE WORD SYMBOL TABLE
                                        ;
001306' 054523  041516          DVIWST: .ASCII  /SYNC/
001312' 000004                          .WORD   DEVIW1
001314' 041523  052116                  .ASCII  /SCNT/
001320' 000006                          .WORD   DEVIW2
001322' 177777                          .WORD   177777                  ;END OF TABLE

                                        ;
                                        ;       MODEL STATEMENT TABLE EXTENSION
                                        ;
001324'    377                  MSFMT3: .BYTE   377
001325'    101     124                  .ASCII  /AT/
001327'    377                  MSFMT2: .BYTE   377
001330'    000                  MSFMT1: .BYTE   0
        001332'                         .EVEN


                                        ;**** CONSTANTS & WORK AREAS ****


        001332'                 HSKPST= .
001332'                         RISTAT:
001332' 000000                  RIRCSR: .WORD   0                       ;STORAGE FOR DEV REG'S ON READ INT
001334' 000000                  RIRBUF: .WORD   0
001336' 000000                  RITCSR: .WORD   0

001340'                         WISTAT:
001340' 000000                  WIRCSR: .WORD   0                       ;STORAGE FOR DEV REG'S ON WRITE INT
001342' 000000                          .WORD   0
001344' 000000                  WITCSR: .WORD   0

001346' 000003                  CSTAT:  .BLKW   3                       ;DEV REG CURRENT VALUES STORAGE

001354' 000000                  OBJADR: .WORD   0                       ;ADR OF CURRENT USER STMNT
001356' 000000                  RDADR:  .WORD   0                       ;CURR DATA ADR FOR READ
001360' 000000                  RDBCNT: .WORD   0                       ;CURR BYTE CNT FOR READ
001362' 000000                  WRADR:  .WORD   0                       ;CURR DATA ADR FOR WRITE
001364' 000000                  WRBCNT: .WORD   0                       ;CURR BYTE CNT FOR WRITE
001366' 000000                  RDSIZE: .WORD   0                       ;# OF BYTES TRANSFERRED ON READ
001370' 000000                  WRSIZE: .WORD   0                       ;# OF BYTE TRANSFERRED ON WRITE
001372' 000000                  TOCNT:  .WORD   0                       ;# OF ENTRIES INTO TIMEOUT ROUT
001374' 177400                  CMASK:  .WORD   177400                  ;PARITY BIT CLEAR MASK
001376' 000000                  PARB:   .WORD   0                       ;PARITY BIT CURR POSITION
001400' 000000                  ISCNT:  .WORD   0                       ;INTERRUPT'S CURR SYNC CHAR CNT
001402' 000000                  PADCNT: .WORD   0                       ;WRITE'S PAD CHAR CNT
001404'                         COUNTS:
001404' 000000                  BYRD:   .WORD   0                       ;BYTES READ COUNT (READ)
001406' 000000                          .WORD   0
001410' 000000                  BYWR:   .WORD   0                       ;BYTES WRITTEN COUNT (WRITE)
001412' 000000                          .WORD   0
001414' 000000                  RDCNT:  .WORD   0                       ;READ CMND COUNT (READ)
001416' 000000                  WRCNT:  .WORD   0                       ;WRITE CMND COUNT (WRITE)
001420' 000000                  BRKCNT: .WORD   0                       ;BREAK CMND COUNT (BREAK)
```

```
383  001422' 000000          MISCNT: .WORD   0                       ;MISC. CMND COUNT
384  001424' 000000          PARCNT: .WORD   0                       ;PARITY ERRORS COUNT
385  001426' 000000          FRMCNT: .WORD   0                       ;FRAMING ERRORS COUNT
386  001430' 000000          OVRCNT: .WORD   0                       ;OVERRUN ERRORS COUNT
387  001432' 000000          DSCCNT: .WORD   0                       ;DATA SET CHANGE ERRORS COUNT
388  001434' 000000          DNACNT: .WORD   0                       ;DATA NOT AVAILABLE ERRORS COUNT
389  001436' 000000          TOECNT: .WORD   0                       ;TIMEOUT ERRORS COUNT
390  001440' 000000          DATAER: .WORD   0                       ;DATA ERRORS COUNT
391  001442' 000000          RDICNT: .WORD   0                       ;READ INTERRUPTS COUNT
392  001444' 000000          WRICNT: .WORD   0                       ;WRITE INTERRUPTS COUNT
393
394          001446'         HSKPEN= .
395
396          000000          XXXX=   0                               ;VALUE TO BE TAILORED BY DEV ROUT
397
398          000200          CACHE=  200                             ;SYSTEM FLDWD BIT DEF.
399
400          000021          CNTNUM= HSKPEN-COUNTS/2                 ;# OF STATISTICAL COUNT WORDS
401
402          000026          SYNCB=  026                             ;PRESET SYNC CHARACTER
403
404          000005          NSYNC=  5                               ;PRESET # OF SYNC CHARACTERS
405
406  001446' 037026          PCSRV:  .WORD   ISYNCM+BITS8+PARENB+SYNCB  ;PCSR REGISTER BASE VALUE
407
408
409  001450'                 PATCH:  .REPT   20.                      ;PATCH AREA
410                                  .WORD   0
411                          .ENDR
```

```
                                .SBTTL  DEVICE REGISTER NAME EQUATES

                                ;DEVICE REGISTER NAME EQUATES RELATIVE TO RCSR AND
                                ;BIT NAME EQUATES FOR THEIR BITS.

000000                  RCSR=   0                       ;RECEIVER CONTROL & STATUS

100000                          DSC     =  100000               ;DATA SET CHANGE
040000                          RING    =  40000                ;RING
020000                          CTS     =  20000                ;CLEAR TO SEND
010000                          CARIER  =  10000                ;CARRIER
001000                          DSR     =  1000                 ;DATA SET READY
000400                          STRSYC  =  400                  ;STRIP SYNC
000100                          RINTEN  =  100                  ;RECEIVER INTERRUPT ENABLE
000040                          DSCIE   =  40                   ;DATA SET CHANGE INT ENABLE
000020                          SCHSYC  =  20                   ;SEARCH SYNC
000004                          RQTS    =  4                    ;REQUEST TO SEND
000002                          DTR     =  2                    ;DATA TERMINAL READY

000002                  RBUF=   2                       ;RECEIVER DATA BUFFER

100000                          RXER    =  100000               ;RECEIVER ERROR
040000                          OVR     =  40000                ;OVERRUN ERROR
020000                          FRM     =  20000                ;FRAMING ERROR
010000                          PAR     =  10000                ;PARITY ERROR

000002                  PCSR=   2                       ;PARAMETER CONTROL

030000                          MODEBT  =  30000                ;MODE BITS
030000                          ISYNCM  =  30000                ;INTERNAL SYNCHRONOUS MODE
020000                          ESYNCM  =  20000                ;EXTERNAL SYNCHRONOUS MODE
000000                          ISOCHM  =  00000                ;ISOCHRONOUS MODE
006000                          WDLENG  =  6000                 ;WORD LENGTH BITS
006000                          BITS8   =  6000                 ;8 BITS
004000                          BITS7   =  4000                 ;7 BITS
002000                          BITS6   =  2000                 ;6 BITS
000000                          BITS5   =  0000                 ;5 BITS
001000                          PARENB  =  1000                 ;PARITY ENABLE (0 = NO PARITY)
000400                          PARSEN  =  400                  ;PARITY SENSE (0 = ODD, 1 = EVEN)

000004                  TCSR=   4                       ;TRANSMITTER CONTROL & STATUS

100000                          DNA     =  100000               ;DATA NOT AVAILABLE
014000                          MAINTM  =  14000                ;MAINT. MODE BITS
000400                          MSTRST  =  400                  ;MASTER RESET
000100                          TINTEN  =  100                  ;TRANSMITTER INTERRUPT ENABLE
000040                          TDNAIE  =  40                   ;DNA INTERRUPT ENABLE
000020                          TSEND   =  20                   ;TRANSMITTER SEND
000010                          HLFDPX  =  10                   ;HALF DUPLEX
000001                          BRK     =  1                    ;BREAK

000006                  TBUF=   6                       ;TRANSMITER DATA BUFFER
```

```
                                        .SBTTL  DU11 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

001520' 010700                          HSKEEP: MOV     PC,R0           ;SET UP FIRST WD ADR
001522' 062700  177610                          ADD     #HSKPST-.,R0
001526' 012701  000046                          MOV     #HSKPEN-HSKPST/2,R1     ;SET UP # OF WORDS
001532' 005725                          TST     (R5)+           ;UNCONDITIONALLY DO HSKP?
001534' 001005                          BNE     10$             ;N,Y-10$
001536' 032702  000004                  BIT     #HSKPEP,R2      ;OPSW SPECIFY DON'T HSKP COUNTS?
001542' 001402                          BEQ     10$             ;Y,N-10$
001544' 162701  000021                  SUB     #CNTNUM,R1      ;REMOVE THEM FROM LOOP COUNT
001550' 005020                  10$:    CLR     (R0)+           ;HSKP ALL NECESSARY AREAS
001552' 005301                          DEC     R1
001554' 001375                          BNE     10$
001556' 012767  000026  176220  ICONS:  MOV     #SYNCB,SYNC             ;INITIALIZE SYNC CHARACTER
001564' 012767  000005  176214          MOV     #NSYNC,SCNT             ;INITIALIZE SYNC CHAR COUNT
001572' 012767  037026  177646          MOV     #ISYNCM+BITS8+PARENB+SYNCB,PCSRV  ;INITIALIZE PCSR REG VALUE
001600' 012767  177400  177566          MOV     #177400,CMASK           ;SET PARITY CLEAR MASK TO 8 BITS
001606' 005067  177564                  CLR     PARB            ;SET PARITY BIT POSITION TO 8 BITS
001612' 000205                          RTS     R5              ;EXIT IN-LINE


                                        ;DU11 REPORT ROUTINE (ALSO "STATUS" AND "COUNTS")

                                                ;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

001614' 004067  004116          REPORT: JSR     R0,SAVREG               ;SAVE REG'S R0 - R5
001620' 004767  004144                  JSR     PC,SUPTAD               ;SET UP PROG TBL ADR IN R3
001624' 011504                          MOV     (R5),R4                 ;GET FLAGWORD
001626' 032704  000002                  BIT     #2,R4                   ;GOING TO DO STATUS DISPLAY?
001632' 001403                          BEQ     5$                      ;Y,N-5$
001634' 004567  004150                  JSR     R5,STSTAT               ;GO STORE STATUS REG'S
001640' 177506                          .WORD   CSTAT-.
001642' 032704  177776          5$:     BIT     #177776,R4              ;DISPLAYING CNTS AT END OF
001646' 001012                          BNE     15$                     ;PROG PASS? (Y,N-15$)
001650' 010700                          MOV     PC,R0                   ;SET UP ADR OF CNTS
001652' 062700  177532                  ADD     #COUNTS-.,R0
001656' 012701  000021                  MOV     #CNTNUM,R1              ;GET # OF CNT WORDS
001662' 005720                  10$:    TST     (R0)+                   ;THIS CNT WORD = 0?
001664' 001003                          BNE     15$                     ;Y,N-15$
001666' 005301                          DEC     R1                      ;DECR WORD CNT
001670' 001374                          BNE     10$                     ;CK'ED ALL WORDS? (Y,N-10$)
```

```
000512                          BR      DVREX           ;GO TO EXIT -- ALL CNTS ARE 0'S
004767  004130          15$:    JSR     PC,DISUNM       ;DISPLAY DEVICE I.D.
032704  000002                  BIT     #2,R4           ;DOING STATUS DISPLAY?
001445                          BEQ     DISCNT          ;Y,N-DISCNT
010700                          MOV     PC,R0           ;SET UP ADR OF REG'S AT
062700  177422                  ADD     #RISTAT-.,R0    ;LAST READ INT
012701  000003                  MOV     #3,R1           ;SET UP # OF REG'S
005720                  20$:    TST     (R0)+           ;ALL REG'S = 0?
001003                          BNE     30$             ;N,Y-40$
005301                          DEC     R1
001374                          BNE     20$
000407                          BR      40$
004567  004220          30$:    JSR     R5,PRINT        ;ISSUE 'AT LAST READ INT' MSG
004341                          .WORD   LRIMSG-.
000021                          .WORD   17.
004567  004106                  JSR     R5,DISPST       ;DISPLAY STATUS AT LAST READ INT
177364                          .WORD   RISTAT-.
005767  177364          40$:    TST     WIRCSR          ;WR INT REG STORAGE = 0'S?
001003                          BNE     60$             ;N,Y-70$
005767  177362                  TST     WITCSR
001407                          BEQ     70$
004567  004166          60$:    JSR     R5,PRINT        ;ISSUE 'AT LAST WRITE INT' MSG
004330                          .WORD   LWIMSG-.
000022                          .WORD   18.
004567  004054                  JSR     R5,DISPST       ;DISPLAY STATUS AT LAST WRITE INT
177340                          .WORD   WISTAT-.
004567  004150          70$:    JSR     R5,PRINT        ;ISSUE 'CURRENTLY' MSG
004334                          .WORD   CURMSG-.
000012                          .WORD   10.
004567  004036                  JSR     R5,DISPST       ;DISPLAY CURRENT STATUS
177330                          .WORD   CSTAT-.
032704  000001          DISCNT: BIT     #1,R4           ;DISPLAY COUNTS?
001431                          BEQ     RPTEND          ;Y,N-RPTEND
012700  000021                  MOV     #CNTNUM,R0      ;SET UP # OF WORDS
010701                          MOV     PC,R1           ;SET UP ADR OF CNTS
062701  177350                  ADD     #COUNTS-.,R1
010702                          MOV     PC,R2           ;SET UP TBL ADR
062702  000066                  ADD     #REPTBL-.,R2
012267  000012          RPTLP:  MOV     (R2)+,RPTBAS    ;MOV MSG ADR TO S/R LINKAGE
004067  003665                  JSR     R0,SAVREG       ;SAVE ALL REG'S
011100                          MOV     (R1),R0         ;GET CURRENT COUNT
004577  175772                  JSR     R5,@BINASC      ;CONVERT IT TO ASCII
000000          RPTBAS: .WORD   XXXX
004067  003660                  JSR     R0,RESREG       ;RESTORE REG'S
005721                          TST     (R1)+           ;POINT AT NXT CNT
005300                          DEC     R0              ;DONE ALL WORDS?
001363                          BNE     RPTLP           ;Y,N-RPTLP
004567  004052                  JSR     R5,PRINT        ;GO ISSUE COUNTS MSG
004324                          .WORD   CNTSMG-.
000360                          .WORD   CNTSEN-CNTSMG
004567  004042          RPTEND: JSR     R5,PRINT        ;ISSUE "END OF REPORT" MSG
004240                          .WORD   RENDMG-.
177763                          .WORD   -13.
004067  003626          DVREX:  JSR     R0,RESREG       ;RESTORE REGISTERS
005725                          TST     (R5)+           ;SET UP RETURN POINT
000205                          RTS     R5              ;EXIT IN-LINE
```

```
                                REPTBL: .WORD   BCMRD-RPTBAS
                                        .WORD   BCMRD+6-RPTBAS
                                        .WORD   BCMWR-RPTBAS
                                        .WORD   BCMWR+6-RPTBAS
                                        .WORD   CMDCRD-RPTBAS
                                        .WORD   CMDCWR-RPTBAS
                                        .WORD   CMDBRK-RPTBAS
                                        .WORD   CMDCMS-RPTBAS
                                        .WORD   CNTPAR-RPTBAS
                                        .WORD   CNTFRM-RPTBAS
                                        .WORD   CNTOVR-RPTBAS
                                        .WORD   CNTDSC-RPTBAS
                                        .WORD   CNTDNA-RPTBAS
                                        .WORD   CNTTOE-RPTBAS
                                        .WORD   CNTDER-RPTBAS
                                        .WORD   CNTRDI-RPTBAS
                                        .WORD   CNTWRI-RPTBAS


                                ;TIMEOUT ERROR ROUTINE
                                        ;JSR    R5,TOUTER       S/R CALL

                        TOUTER: INC     TOCNT                   ;ADD 1 TO TIMEOUT CNTR
                                CMP     TOCNT,#8.               ;EIGHTH TIME THRU ON THIS I/O?
                                BEQ     10$                     ;N,Y-10$
                                RTS     R5                      ;EXIT BACK TO MPG
                        10$:    JSR     R0,SAVREG               ;SAVE ALL REGISTERS
                                INC     TOECNT                  ;ADD 1 TO TIMEOUT ERROR CNTR
                                JSR     PC,SUPTAD               ;SET UP RCSR & PROG TBL ADR'S
                                JSR     R5,STSTAT               ;STORE CURRENT STATUS
                                .WORD   CSTAT-.
                                JSR     R5,KILL                 ;RESET I.E.'S & VECTORS
                                BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O FLAG
                                JSR     R5,ERRCS                ;ISSUE TIMEOUT ERROR MSG
                                .WORD   IOTO-ERMBAS
                                JSR     R0,RESREG               ;RESTORE REGISTERS
                                MOV     (SP)+,R5                ;REMOVE RETURN ADR
                                JMP     @CUPGER                 ;GO TO ERROR EXIT
```

```
                                        ;KILL USER PROGRAM ROUTINE
                                        ;
                                        ;JSR    R5,KILL          S/R CALL
                                        ;
                                        ;R3 = PROG TBL ADR
                                        ;
                                        ;DESTROYS R0,R1
002262' 016701 175536          KILL:    MOV    DREGAD,R1                ;GET RCSR ADR
002266' 032711 000140                   BIT    #RINTEN+DSCIE,(R1)       ;READ INT ENABLES SET?
002272' 001407                          BEQ    10$                      ;Y,N-10$
002274' 004567 002674                   JSR    R5,TRVECT                ;IS READ VECTOR SET FOR ME?
002300' 000404                          BR     10$                      ;Y,N-10$
002302' 042711 000160                   BIC    #RINTEN+SCHSYC+DSCIE,(R1)  ;RESET RD INT ENABLES & SRCH SYNC
002306' 004767 002604                   JSR    PC,RRINTV                ;RESET READ'S INT VECTOR
002312' 032761 000140 000004   10$:     BIT    #TINTEN+TDNAIE,TCSR(R1)  ;WRITE INT ENABLES SET?
002320' 001410                          BEQ    20$                      ;Y,N-20$
002322' 004567 002676                   JSR    R5,TWVECT                ;IS WRITE VECTOR SET FOR ME?
002326' 000405                          BR     20$                      ;Y,N-20$
002330' 042761 000140 000004            BIC    #TINTEN+TDNAIE,TCSR(R1)  ;RESET WR INT ENABLES
002336' 004767 002600                   JSR    PC,RWINTV                ;RESET WRITE'S INT VECTOR
002342' 042767 000003 175432   20$:     BIC    #RDBSY+WRBSY,DFLGWD      ;RESET THE BUSY FLAGS
002350' 000205                          RTS    R5                       ;EXIT IN-LINE
```

```
                                        .SBTTL  DU11 NON I/O FUNCTION ROUTINES


                                        ;"WAIT" FUNCTION ROUTINE

                                                ;JSR     R5,WAIT         FUNCTION CALL

002352' 042767  100000  175422  WAIT:   BIC     #WAITMD,DFLGWD          ;RESET THE "NOWAIT" FLAG
002360' 004767  003404                  JSR     PC,SUPTAD               ;GET PROG TBL ADR IN R3
002364' 052713  000010                  BIS     #WT4IOT,(R3)            ;SET WAITING FOR I/O TERM
002370' 032767  000003  175404          BIT     #RDBSY+WRBSY,DFLGWD     ;READ OR WRITE STILL BUSY?
002376' 001402                          BEQ     10$                     ;Y,N-10$
002400' 004577  175442                  JSR     R5,@CIOBSY              ;WAIT FOR I/O TO COMPLETE
002404' 042713  000010          10$:    BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
002410' 004767  002206                  JSR     PC,CKBSY                ;WAIT IF DU11 IS BUSY & DO TERMINATION
002414' 000205                          RTS     R5                      ;EXIT IN-LINE


                                        ;"NOWAIT" FUNCTION ROUTINE

                                                ;JSR     R5,NOWAIT       FUNCTION CALL

002416' 052767  100000  175356  NOWAIT: BIS     #WAITMD,DFLGWD          ;SET THE "NOWAIT" FLAG
002424' 000205                          RTS     R5                      ;EXIT IN-LINE


                                        ;"CVSYNC" FUNCTION ROUTINE

                                                ;JSR     R5,CVSYNC       FUNCTION CALL
                                                ;.WORD   ADR             DATA ADDRESS
                                                ;.WORD   CNT             BYTE COUNT

002426' 012500          CVSYNC: MOV     (R5)+,R0                        ;GET THE DATA ADDRESS
002430' 012501                  MOV     (R5)+,R1                        ;GET THE BYTE COUNT
002432' 016703  175346          MOV     SYNC,R3                         ;GET THE SYNC CHAR
002436' 046703  176732          BIC     CMASK,R3                        ;SAVE ONLY PERTINENT BITS
002442' 112002          10$:    MOVB    (R0)+,R2                        ;GET THE DATA BYTE
002444' 046702  176724          BIC     CMASK,R2                        ;SAVE ONLY PERTINENT BITS
002450' 020203                  CMP     R2,R3                           ;DATA BYTE = SYNC BYTE?
002452' 001002                  BNE     20$                             ;Y,N-20$
002454' 105160  177777          COMB    -1(R0)                          ;COMPLEMENT THE DATA BYTE
002460' 005301          20$:    DEC     R1                              ;DECR THE BYTE COUNT
002462' 001367                  BNE     10$                             ;DONE ALL BYTES? (Y,N-10$)
002464' 000205                  RTS     R5                              ;EXIT TO USER PROG
```

```
                                        ;"GENPAR" FUNCTION ROUTINE
                                        ;JSR    R5,GENPAR       FUNCTION CALL
                                        ;.WORD  ADR             DATA ADDRESS
                                        ;.WORD  CNT             BYTE COUNT

002466' 012500          GENPAR: MOV     (R5)+,R0                ;GET THE DATA ADDRESS
002470' 012501                  MOV     (R5)+,R1                ;GET THE BYTE COUNT
002472' 111002          10$:    MOVB    (R0),R2                 ;GET THE DATA BYTE
002474' 046702 176674           BIC     CMASK,R2                ;SAVE ONLY PERTINENT BITS
002500' 032767 001000 176740    BIT     #PARENB,PCSRV           ;IS PARITY BEING USED?
002506' 001426                  BEQ     50$                     ;Y,N-50$
002510' 010204                  MOV     R2,R4                   ;MOVE BYTE TO WORK REG
002512' 005003                  CLR     R3                      ;CLEAR ACCUMULATOR REG
002514' 006204          20$:    ASR     R4                      ;SHIFT OUT LSB
002516' 001402                  BEQ     30$                     ;ANY BITS LEFT? (Y,N-30$)
002520' 005503                  ADC     R3                      ;ADD CARRY BIT TO BIT CNT
002522' 000774                  BR      20$                     ;GO SHIFT SOME MORE
002524' 005503          30$:    ADC     R3                      ;DON'T FORGET LAST CARRY
002526' 032767 000400 176712    BIT     #PARSEN,PCSRV           ;PARITY = ODD PARITY?
002534' 001006                  BNE     40$                     ;Y,N-40$
002536' 032703 000001           BIT     #1,R3                   ;ODD # OF BITS?
002542' 001010                  BNE     50$                     ;N,Y-50$
002544' 056702 176626           BIS     PARB,R2                 ;SET IN THE PARITY BIT
002550' 000405                  BR      50$                     ;GO STORE THE BYTE
002552' 032703 000001   40$:    BIT     #1,R3                   ;EVEN # OF BITS?
002556' 001402                  BEQ     50$                     ;N,Y-50$
002560' 056702 176612           BIS     PARB,R2                 ;SET IN THE PARITY BIT
002564' 110220          50$:    MOVB    R2,(R0)+                ;STORE THE MODIFIED DATA BYTE
002566' 005301                  DEC     R1                      ;DECR BYTE COUNT
002570' 001340                  BNE     10$                     ;DONE ALL BYTES? (Y,N-10$)
002572' 000205                  RTS     R5                      ;EXIT TO USER PROG
```

```
                                        .SBTTL  DU11 NON-INTERRUPT I/O FUNCTION ROUTINES


                                        ;"CALL" AND "ANSWER" FUNCTION ROUTINES

                                                ;JSR    R5,CALL         FUNCTION CALL
                                                ;JSR    R5,ANSWER       FUNCTION CALL

002574'                         CALL:
002574' 052777  000002  175222  ANSWER: BIS     #DTR,@DREGAD            ;SET DATA TERMINAL RDY IN RCSR
002602' 032777  001000  175214  10$:    BIT     #DSR,@DREGAD            ;DATA SET RDY = 1?
002610' 001003                          BNE     NINTEX                  ;N,Y-NINTEX
002612' 004577  175230                  JSR     R5,@CIOBSY              ;RELEASE CONTROL FOR 1 POLLING LOOP
002616' 000771                          BR      10$                     ;GO CK FOR DATA SET RDY AGAIN
002620' 005267  176576          NINTEX: INC     MISCNT                  ;ADD 1 TO MISC. CMND COUNT
002624' 000205                          RTS     R5                      ;EXIT TO USER PROG


                                        ;"LISTEN" FUNCTION ROUTINE

                                                ;JSR    R5,LISTEN       FUNCTION CALL

002626' 032777  040000  175170  LISTEN: BIT     #RING,@DREGAD           ;IS THE RING BIT SET?
002634' 001371                          BNE     NINTEX                  ;N,Y-NINTEX
002636' 004577  175204                  JSR     R5,@CIOBSY              ;RELEASE CONTROL FOR 1 POLLING LOOP
002642' 000771                          BR      LISTEN                  ;GO CK RING AGAIN


                                        ;"READY" FUNCTION ROUTINE

                                                ;JSR    R5,READY        FUNCTION CALL

002644' 032777  010000  175152  READY:  BIT     #CARIER,@DREGAD         ;IS THE CARRIER UP YET?
002652' 001362                          BNE     NINTEX                  ;N,Y-NINTEX
002654' 004577  175166                  JSR     R5,@CIOBSY              ;RELEASE CONTROL FOR 1 POLLING LOOP
002660' 000771                          BR      READY                   ;GO CK CARRIER AGAIN


                                        ;"RECV" FUNCTION ROUTINE

                                                ;JSR    R5,RECV         FUNCTION CALL

002662' 042777  000004  175134  RECV:   BIC     #RQTS,@DREGAD           ;RESET REQUEST TO SEND IN RCSR
002670' 000753                          BR      NINTEX                  ;GO TO EXIT


                                        ;"SEND" FUNCTION ROUTINE

                                                ;JSR    R5,SEND         FUNCTION CALL

002672' 052777  000004  175124  SEND:   BIS     #RQTS,@DREGAD           ;SET REQUEST TO SEND IN RCSR
002700' 032777  020000  175116  70$:    BIT     #CTS,@DREGAD            ;IS CLEAR TO SEND SET?
002706' 001344                          BNE     NINTEX                  ;N,Y-NINTEX
002710' 004577  175132                  JSR     R5,@CIOBSY              ;RELEASE CONTROL FOR 1 POLLING LOOP
002714' 000771                          BR      70$                     ;GO CK CLEAR TO SEND AGAIN
```

```
                                        ;"HANGUP" FUNCTION ROUTINE

                                                ;JSR     R5,HANGUP       FUNCTION CALL

002716' 042777  000004  175100  HANGUP: BIC     #RQTS,@DREGAD           ;RESET REQ TO SEND IN RCSR
002724' 012700  000017                  MOV     #15.,R0                 ;INITIALIZE LOOP COUNT
002730' 012701  000522          50$:    MOV     #522,R1                 ;INIT 1 MS COUNTER
002734' 032777  000200  175122          BIT     #CACHE,@CSYSFW          ;CPU HAVE A CACHE MEM?
002742' 001402                          BEQ     60$                     ;Y,N-60$
002744' 012701  002126                  MOV     #2126,R1                ;INCREASE 1 MS COUNTER
002750' 005301                  60$:    DEC     R1                      ;DECR 1 MS COUNTER
002752' 001376                          BNE     60$                     ;EXHAUSTED CNTR? (Y,N-60$)
002754' 005300                          DEC     R0                      ;DECR LOOP COUNT
002756' 001364                          BNE     50$                     ;CNT = 0? (Y,N-50$)
002760' 042777  000002  175036          BIC     #DTR,@DREGAD            ;LOWER DATA TERMINAL RDY
002766' 000714                          BR      NINTEX                  ;GO TO EXIT


                                        ;"CRESET" FUNCTION ROUTINE

                                                ;JSR     R5,CRESET       FUNCTION CALL

002770' 016704  175030          CRESET: MOV     DREGAD,R4               ;GET RCSR ADR
002774' 052764  000400  000004          BIS     #MSTRST,TCSR(R4)        ;SET THE MASTER CLEAR BIT
003002' 012700  000024                  MOV     #20.,R0                 ;SET UP DELAY COUNT
003006' 005300                  10$:    DEC     R0                      ;DELAY FOR A FEW
003010' 001376                          BNE     10$                     ;MICROSECONDS
003012' 000205                          RTS     R5                      ;EXIT TO USER PROG


                                        ;"STRIP" FUNCTION ROUTINE

                                                ;JSR     R5,STRIP        FUNCTION CALL

003014' 052777  000400  175002  STRIP:  BIS     #STRSYC,@DREGAD         ;SET STRIP SYNC IN RCSR
003022' 000205                          RTS     R5                      ;EXIT TO USER PROG


                                        ;"NSTRIP" FUNCTION ROUTINE

                                                ;JSR     R5,NSTRIP       FUNCTION CALL

003024' 042777  000400  174772  NSTRIP: BIC     #STRSYC,@DREGAD         ;RESET STRIP SYNC BIT IN RCSR
003032' 000205                          RTS     R5                      ;EXIT TO USER PROG


                                        ;"FDUPLX" FUNCTION ROUTINE

                                                ;JSR     R5,FDUPLX       FUNCTION CALL

003034' 016704  174764          FDUPLX: MOV     DREGAD,R4               ;GET RCSR ADR
003040' 042764  000010  000004          BIC     #HLFDPX,TCSR(R4)        ;RESET THE HALF DUPLEX BIT IN TCSR
003046' 000205                          RTS     R5                      ;EXIT TO USER PROG
```

```
                                        ;"HDUPLX" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,HDUPLX       FUNCTION CALL
                                        ;
003050' 016704  174750          HDUPLX: MOV     DREGAD,R4               ;GET RCSR ADR
003054' 052764  000010  000004          BIS     #HLFDPX,TCSR(R4)        ;SET THE HALF DUPLEX BIT IN TCSR
003062' 000205                          RTS     R5                      ;EXIT TO USER PROG
                                        ;
                                        ;"NORMAL" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,NORMAL       FUNCTION CALL
                                        ;
003064' 016704  174734          NORMAL: MOV     DREGAD,R4               ;GET RCSR ADR
003070' 042764  014000  000004          BIC     #MAINTM,TCSR(R4)        ;SET MAINT MODE BITS TO NORMAL IN TCSR
003076' 000205                          RTS     R5                      ;EXIT TO USER PROG
                                        ;
                                        ;"SYSTST" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,SYSTST       FUNCTION CALL
                                        ;
003100' 016704  174720          SYSTST: MOV     DREGAD,R4               ;GET RCSR ADR
003104' 052764  014000  000004          BIS     #MAINTM,TCSR(R4)        ;SET MAINT MODE BITS TO SYS TEST IN TCSR
003112' 000205                          RTS     R5                      ;EXIT TO USER PROG
                                        ;
                                        ;"EVEN" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,EVEN         FUNCTION CALL
                                        ;
003114' 052767  001400  176324  EVEN:   BIS     #PARENB+PARSEN,PCSRV    ;SET PAR ENB & PAR SEN SEL IN PCSRV
003122' 000412                          BR      LDPCSR                  ;GO LOAD PCSR REG & EXIT
                                        ;
                                        ;"ODD" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,ODD          FUNCTION CALL
                                        ;
003124' 052767  001000  176314  ODD:    BIS     #PARENB,PCSRV           ;SET PARITY ENABLE IN PCSR BASE
003132' 042767  000400  176306          BIC     #PARSEN,PCSRV           ;RESET PAR SEN SEL TO ODD
003140' 000403                          BR      LDPCSR                  ;GO LOAD PCSR REG & EXIT
                                        ;
                                        ;"NOPAR" FUNCTION ROUTINE
                                        ;
                                                ;JSR     R5,NOPAR        FUNCTION CALL
                                        ;
003142' 042767  001400  176276  NOPAR:  BIC     #PARENB+PARSEN,PCSRV    ;RESET PARITY BITS IN PCSR BASE
003150' 116767  174630  176270  LDPCSR: MOVB    SYNC,PCSRV              ;MOVE CURR SYNC CHAR TO PCSR BASE
003156' 016704  174642                  MOV     DREGAD,R4               ;GET RCSR ADR
003162' 016764  176260  000002          MOV     PCSRV,PCSR(R4)          ;LOAD PCSR WITH NEW BASE VALUE
003170' 012700  000310                  MOV     #200.,R0                ;SET UP LOOP CNT
003174' 005300                  10$:    DEC     R0                      ;DELAY A
003176' 001376                          BNE     10$                     ;BIT
003200' 000205                          RTS     R5                      ;EXIT TO USER PROG
```

```
                                        ;"MODE" FUNCTION ROUTINE
                                        ;
                                        ;       JSR     R5,MODE         FUNCTION CALL
                                        ;       .WORD   CODE            MODE SELECT CODE (0, 2, OR 3)
                                        ;
003202' 004767  001502          MODE:   JSR     PC,STMADR               ;STORE USER STMNT ADR
003206' 012500                  MOV     (R5)+,R0                ;GET MODE SELECT CODE
003210' 012701  030000          MOV     #ISYNCM,R1              ;SET UP FOR MODE 3 (INT. SYNC)
003214' 020027  000003          CMP     R0,#3                   ;CODE TOO HIGH?
003220' 101016                  BHI     MODERR                  ;N,Y-MODERR
003222' 001407                  BEQ     MODCOM                  ;IS IT A 3? (N,Y-MODCOM)
003224' 012701  020000          MOV     #ESYNCM,R1              ;SET UP FOR MODE 2 (EXT. SYNC)
003230' 020027  000001          CMP     R0,#1                   ;IS IT A 1?
003234' 001410                  BEQ     MODERR                  ;N,Y-MODERR
003236' 101001                  BHI     MODCOM                  ;IS IT A 2? (N,Y-MODCOM)
003240' 005001                  CLR     R1                      ;SET UP MODE 0 (ISOCHRONOUS)
003242' 042767  030000  176176  MODCOM: BIC     #MODEBT,PCSRV           ;RESET MODE BITS IN PCSR BASE
003250' 050167  176172          BIS     R1,PCSRV                ;SET IN NEW MODE SELECT BITS
003254' 000735                  BR      LDPCSR                  ;GO LOAD PCSR REG & EXIT
                                ;
003256' 004567  002000          MODERR: JSR     R5,ERRCS                ;REPORT INVALID MODE SELECT CODE
003262' 001577                  .WORD   IVMODE-ERMBAS
003264' 000434                  BR      NIEREX                  ;GO TO ERROR EXIT
                                ;
                                ;
                                        ;"BITS" FUNCTION ROUTINE
                                        ;
                                        ;       JSR     R5,BITS         FUNCTION CALL
                                        ;       .WORD   CODE            BITS SELECT CODE (5 - 8)
                                        ;
003266' 004767  001416          BITS:   JSR     PC,STMADR               ;STORE USER STMNT ADR
003272' 012500                  MOV     (R5)+,R0                ;GET THE BITS SELECT CODE
003274' 162700  000005          SUB     #5,R0                   ;ADJ IT DOWNWARD
003300' 020027  000003          CMP     R0,#3                   ;IS IT A VALID CODE?
003304' 101021                  BHI     BITERR                  ;Y,N-BITERR
003306' 006300                  ASL     R0                      ;CONVERT IT
003310' 010046                  MOV     R0,-(SP)                ;TO A
003312' 006300                  ASL     R0                      ;TABLE
003314' 062600                  ADD     (SP)+,R0                ;DISPLACEMENT
003316' 060700                  ADD     PC,R0                   ;ADD IN THE START ADR OF
003320' 062700  000046          ADD     #BITTBL-.,R0            ;THE BIT CODE TABLE
003324' 042767  006000  176114  BIC     #WDLENG,PCSRV           ;RESET CURR BITS IN PCSR BASE
003332' 052067  176110          BIS     (R0)+,PCSRV             ;SET IN THE NEW BITS
003336' 012067  176032          MOV     (R0)+,CMASK             ;STORE THIS CODE'S PARITY CLR MASK
003342' 011067  176030          MOV     (R0),PARB               ;STORE ITS PARITY BIT POSITION
003346' 000700                  BR      LDPCSR                  ;GO LOAD PCSR REG & EXIT
                                ;
003350' 004567  001706          BITERR: JSR     R5,ERRCS                ;REPORT INV BIT SELECT CODE
003354' 001622                  .WORD   IVBITS-ERMBAS
003356' 005267  176056          NIEREX: INC     DATAER                  ;ADD 1 TO DATA ERROR CNT
003362' 000177  174462          JMP     @CUPGER                 ;GO TO MPG'S ERR RETURN POINT
                                ;
003366' 000000  177740  000040  BITTBL: .WORD   BITS5,177740,000040       ;5 BITS - PCSR BITS, BIT CLR MASK,
003374' 002000  177700  000100          .WORD   BITS6,177700,000100     ;6 BITS   PARITY BIT POSITION
003402' 004000  177600  000200          .WORD   BITS7,177600,000200     ;7 BITS
003410' 006000  177400  000000          .WORD   BITS8,177400,000000     ;8 BITS
```

```
                                        ;"PRESET" FUNCTION ROUTINE
                                        ;JSR    R5,PRESET       FUNCTION CALL
003416' 004567 176134          PRESET: JSR     R5,ICONS                ;GO INITIALIZE ALL CONSTANTS
003422' 016704 174376                  MOV     DREGAD,R4               ;GET RCSR ADR
003426' 042764 014011 000004           BIC     #MAINTM+HLFDPX+BRK,TCSR(R4)  ;SET NORM MODE, FULL DUPLX IN TCSR
003434' 004567 177510                  JSR     R5,LDPCSR               ;LOAD THE PCSR REG
003440' 052714 000400                  BIS     #STRSYC,(R4)            ;SET STRIP SYNC BIT IN RCSR
003444' 000205                         RTS     R5                      ;EXIT TO USER PROG
```

```
                                        .SBTTL  DU11 INTERRUPT TYPE I/O FUNCTION ROUTINES


                                        ;"READ" FUNCTION ROUTINE

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

003446' 004767  001134          READ:   JSR     PC,CKRBSY               ;GO CK IF READ IS BUSY
003452' 005725                          TST     (R5)+                   ;BYPASS MSW OF ADR
003454' 012567  175676                  MOV     (R5)+,RDADR             ;STORE THE READ DATA ADR
003460' 012567  175674                  MOV     (R5)+,RDBCNT            ;STORE THE READ BYTE COUNT
003464' 005725                          TST     (R5)+                   ;BYPASS UNUSED WORD
003466' 005267  175722                  INC     RDCNT                   ;ADD 1 TO READ CMND COUNT
003472' 005067  174324                  CLR     ERRI                    ;RESET THE ERROR INDICATOR
003476' 042767  000064  174276          BIC     #RDTERM+RDIERR+RDIDSC,DFLGWD  ;HSKP INTERRUPT FLAGS
003504' 005063  000030                  CLR     PTOCNT(R3)              ;RESET THE TIMEOUT COUNTERS
003510' 005067  175656                  CLR     TOCNT
003514' 005067  175646                  CLR     RDSIZE                  ;RESET # OF BYTES READ
003520' 016767  174302  000012          MOV     IVCTAD,10$              ;GET READ INT VECT ADR
003526' 016767  174276  000006          MOV     RDPSWD,20$              ;GET READ PSW
003534' 004577  174326                  JSR     R5,@SETVEC              ;GO SET UP READ'S INT VECTOR
003540' 000000                  10$:     .WORD   XXXX                    ;  INT VECTOR ADR
003542' 000000                  20$:     .WORD   XXXX                    ;  PSW
003544' 000314                           .WORD   DURINT-.                ;  REL INT ROUT ADR
003546' 052767  000001  174226          BIS     #RDBSY,DFLGWD           ;SET READ'S BUSY FLAG
003554' 052713  000010                  BIS     #WT4IOT,(R3)            ;SET WAITING FOR I/O TERM
003560' 005764  000002                  TST     RBUF(R4)                ;HSKP ANY ERROR BITS
003564' 052714  000120                  BIS     #SCHSYC+RINTEN,(R4)     ;SET SEARCH SYNC & INT ENABLE
003570' 005767  174206          RDWREX: TST     DFLGWD                  ;"NOWAIT" BIT SET?
003574' 100003                          BPL     30$                     ;Y,N-30$
003576' 042713  000010                  BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
003602' 000404                          BR      40$                     ;GO TO EXIT
003604' 004577  174236          30$:    JSR     R5,@CIOBSY              ;WAIT FOR I/O TO COMPLETE
003610' 004767  001110                  JSR     PC,PROCTM               ;GO PROCESS TERMINATION
003614' 000205                  40$:    RTS     R5                      ;EXIT IN-LINE TO USER PROG
```

```
                                        ;"WRITE" AND "BREAK" FUNCTION ROUTINES
                                        ;
                                        ;JSR     R5,WRITE        FUNCTION CALL
                                        ;JSR     R5,BREAK        FUNCTION CALL
                                        ;.WORD   ADR             (NOT USED)
                                        ;.WORD   ADR             DATA ADDRESS (BITS 0 - 15)
                                        ;.WORD   CNT             BYTE COUNT
                                        ;.WORD   DEV             (NOT USED)
                                        ;
003616' 004767  000772          WRITE:  JSR     PC,CKWBSY       ;GO CK IF WRITE IS BUSY
003622' 005267  175570                  INC     WRCNT           ;ADD 1 TO WRITE CMND COUNT
003626' 042767  000200  174146          BIC     #BRKFLG,DFLGWD  ;RESET THE BREAK FLAG
003634' 000407                          BR      WRBRCM          ;GO TO WRITE/BREAK COMMON POINT
                                        ;
003636' 004767  000752          BREAK:  JSR     PC,CKWBSY       ;GO CK IF WRITE IS BUSY
003642' 005267  175552                  INC     BRKCNT          ;ADD 1 TO BREAK CMND COUNT
003646' 052767  000200  174126          BIS     #BRKFLG,DFLGWD  ;SET THE BREAK FLAG
                                        ;
003654' 004767  000734          WRBRCM: JSR     PC,CKWBSY       ;GO CK IF WRITE IS BUSY
003660' 005725                          TST     (R5)+           ;BYPASS MSW OF ADR
003662' 012567  175474                  MOV     (R5)+,WRADR     ;STORE THE WRITE DATA ADR
003666' 012567  175472                  MOV     (R5)+,WRBCNT    ;STORE THE WRITE BYTE COUNT
003672' 005725                          TST     (R5)+           ;BYPASS UNUSED WORD
003674' 042767  000110  174100          BIC     #WRTERM+WRIERR,DFLGWD   ;HSKP INTERRUPT FLAGS
003702' 005067  174114                  CLR     ERRI            ;CLEAR THE ERROR INDICATOR
003706' 005063  000030                  CLR     PTOCNT(R3)      ;RESET THE TIMEOUT COUNTERS
003712' 005067  175454                  CLR     TOCNT
003716' 005067  175446                  CLR     WRSIZE          ;INITIALIZE # OF BYTES WRITTEN
003722' 005067  175454                  CLR     PADCNT          ;RESET THE PAD CHAR CNT
003726' 016767  174074  000020          MOV     IVCTAD,50$      ;GET INT VECTOR BASE ADR
003734' 062767  000004  000012          ADD     #4,50$          ;POINT IT AT WRITE'S VECTOR
003742' 016767  174064  000006          MOV     WRPSWD,60$      ;GET WRITE PSW
003750' 004577  174112                  JSR     R5,@SETVEC      ;GO SET UP WRITE'S INT VECTOR
003754' 000000                  50$:    .WORD   XXXX            ;  INT VECTOR ADR
003756' 000000                  60$:    .WORD   XXXX            ;  PSW
003760' 000362                          .WORD   DUWINT-.        ;  REL INT ROUT ADR
003762' 005067  175412                  CLR     ISCNT           ;RESET THE SYNC COUNT
003766' 032767  030000  175452          BIT     #MODEBT,PCSRV   ;IN ISOCHRONOUS MODE?
003774' 001403                          BEQ     70$             ;N,Y-70$
003776' 016767  174004  175374          MOV     SCNT,ISCNT      ;INITIALIZE INT'S SYNC CHAR CNT
004004' 042764  000001  000004  70$:    BIC     #BRK,TCSR(R4)   ;RESET THE BREAK BIT IN TCSR
004012' 032767  000200  173762          BIT     #BRKFLG,DFLGWD  ;DOING A BREAK INST?
004020' 001403                          BEQ     80$             ;Y,N-80$
004022' 052764  000001  000004          BIS     #BRK,TCSR(R4)   ;SET THE BREAK BIT IN TCSR
004030' 052764  000020  000004  80$:    BIS     #TSEND,TCSR(R4) ;SET THE SEND BIT
004036' 052767  000002  173736          BIS     #WRBSY,DFLGWD   ;SET WRITE'S BUSY FLAG
004044' 052713  000010                  BIS     #WT4IOT,(R3)    ;SET WAITING FOR I/O TERM
004050' 052764  000140  000004          BIS     #TINTEN+TDNAIE,TCSR(R4) ;SET BOTH INT ENABLES
004056' 000644                          BR      RDWREX          ;GO CK WAIT/NOWAIT FLAG
```

```
                                        .SBTTL  DU11 READ INTERRUPT SERVICE ROUTINE


                                        ;RECEIVER INTERRUPT ENTRY POINT


004060' 004067  001652          DURINT: JSR     R0,SAVREG               ;SAVE REGISTERS R0 THRU R5
004064' 004567  001720                  JSR     R5,STSTAT               ;GO STORE ALL DEV REGS
004070' 175242                          .WORD   RISTAT-.
004072' 005267  175344                  INC     RDICNT                  ;ADD 1 TO READ INTERRUPT COUNT
004076' 004767  001666                  JSR     PC,SUPTAD               ;SET UP PROG TBL & RCSR ADRS
004102' 005767  175260                  TST     RDSIZE                  ;FIRST INT ON READ?
004106' 001005                          BNE     10$                     ;Y,N-10$
004110' 042767  100000  175214          BIC     #DSC,RIRCSR             ;IGNORE DATA SET CHG ON 1ST TIME
004116' 052714  000040                  BIS     #DSCIE,(R4)             ;SET DSC INT ENABLE
004122' 016701  175206          10$:    MOV     RIRBUF,R1               ;GET STORED RBUF REG
004126' 005767  175200                  TST     RIRCSR                  ;DATA SET CHG INT?
004132' 100453                          BMI     DSCERR                  ;N,Y-DSCERR
004134' 032701  170000                  BIT     #RXER+OVR+FRM+PAR,R1    ;ANY ERRORS IN STORED RBUF WORD?
004140' 001055                          BNE     RDERR                   ;N,Y-RDERR
004142' 005767  175212                  TST     RDBCNT                  ;DATA BYTE CNT = 0?
004146' 001417                          BEQ     RDRBSY                  ;N,Y-RDRBSY
004150' 016700  175202                  MOV     RDADR,R0                ;GET CURR RD DATA ADR
004154' 004777  173720                  JSR     PC,@PUTBYT              ;STORE DATA BYTE IN MEM
004160' 010067  175172                  MOV     R0,RDADR                ;STORE NEW RD DATA ADR
004164' 005267  175216                  INC     BYRD+2                  ;ADD 1 TO TOTAL BYTES
004170' 005567  175210                  ADC     BYRD                    ;READ COUNTER
004174' 005267  175166                  INC     RDSIZE                  ;ADD 1 TO THIS MSG'S SIZE
004200' 005367  175154                  DEC     RDBCNT                  ;DECR THE DATA BYTE COUNT
004204' 001022                          BNE     INTEX                   ;BYTE CNT = 0? (Y,N-INTEX)

004206' 042767  000001  173566  RDRBSY: BIC     #RDBSY,DFLGWD           ;RESET READ BUSY FLAG
004214' 052767  000004  173560          BIS     #RDTERM,DFLGWD          ;SET DO READ TERMINATION FLAG
004222' 042714  000160                  BIC     #RINTEN+DSCIE+SCHSYC,(R4)  ;RESET SEARCH SYNC & INT ENB FOR RD
004226' 032764  000100  000004          BIT     #TINTEN,TCSR(R4)        ;IS WRITE INT ENABLE SET?
004234' 001404                          BEQ     CLRWTF                  ;Y,N-CLRWTF
004236' 032767  000002  173536          BIT     #WRBSY,DFLGWD           ;IS WRITE BUSY?
004244' 001002                          BNE     INTEX                   ;N,Y-INTEX
004246' 042713  000010          CLRWTF: BIC     #WT4IOT,(R3)            ;RESET WAITING FOR I/O TERM
004252' 004067  001474          INTEX:  JSR     R0,RESREG               ;RESTORE REGISTERS R0 THRU R5
004256' 000177  173612                  JMP     @RTNINT                 ;EXIT FROM INTERRUPT

004262' 052767  000040  173512  DSCERR: BIS     #RDIDSC,DFLGWD          ;SET DATA SET CHANGE ERROR FLG
004270' 005267  175136                  INC     DSCCNT                  ;ADD 1 TO DSC ERROR CNTR
004274' 052767  000020  173500  RDERR:  BIS     #RDIERR,DFLGWD          ;SET THE READ INTERRUPT ERR FLAG
004302' 032701  040000                  BIT     #OVR,R1                 ;IS THERE AN OVERRUN ERROR?
004306' 001402                          BEQ     RDE1                    ;Y,N-RDE1
004310' 005267  175114                  INC     OVRCNT                  ;ADD 1 TO OVR ERROR CNTR
004314' 032701  020000          RDE1:   BIT     #FRM,R1                 ;IS THERE A FRAMING ERROR?
004320' 001402                          BEQ     RDE2                    ;Y,N-RDE2
004322' 005267  175100                  INC     FRMCNT                  ;ADD 1 TO FRM ERROR CNTR
004326' 032701  010000          RDE2:   BIT     #PAR,R1                 ;IS THERE A PARITY ERROR?
004332' 001725                          BEQ     RDRBSY                  ;Y,N-RDRBSY
004334' 005267  175064                  INC     PARCNT                  ;ADD 1 TO PAR ERROR CNTR
004340' 000722                          BR      RDRBSY                  ;GO TERMINATE THE READ
```

```
                                        .SBTTL  DU11 WRITE INTERRUPT SERVICE ROUTINE


                                        ;TRANSMITTER INTERRUPT ENTRY POINT


004342' 004067  001370          DUWINT: JSR     R0,SAVREG               ;SAVE REGISTERS R0 THRU R5
004346' 005267  175072                  INC     WRICNT                  ;ADD 1 TO WRITE INTERRUPT COUNT
004352' 004767  001412                  JSR     PC,SUPTAD               ;SET UP PROG TBL & RCSR ADRS
004356' 016467  000004  174760          MOV     TCSR(R4),WITCSR         ;STORE CURRENT TCSR CONTENTS
004364' 011467  174750                  MOV     (R4),WIRCSR             ;STORE RCSR ALSO
004370' 005767  174750                  TST     WITCSR                  ;DNA SET IN STORED TCSR?
004374' 100476                          BMI     DNAERR                  ;N,Y-DNAERR
004376' 005767  174776                  TST     ISCNT                   ;SENDING SYNC CHARACTERS?
004402' 001406                          BEQ     WXDATA                  ;Y,N-WXDATA
004404' 116764  173374  000006          MOVB    SYNC,TBUF(R4)           ;LOAD ANOTHER SYNC CHAR
004412' 005367  174762                  DEC     ISCNT                   ;DECR SYNC CHAR CNT
004416' 000423                          BR      WINCBC                  ;GO ADD 1 TO BYTE COUNT

004420' 005767  174740          WXDATA: TST     WRBCNT                  ;ANY MORE DATA BYTES TO SEND?
004424' 001427                          BEQ     WCKPAD                  ;Y,N-WCKPAD
004426' 012701  000377                  MOV     #377,R1                 ;PRESET TO THE BREAK DATA
004432' 032767  000200  173342          BIT     #BRKFLG,DFLGWD          ;DOING A BREAK INST?
004440' 001006                          BNE     10$                     ;N,Y-10$
004442' 016700  174714                  MOV     WRADR,R0                ;GET CURRENT DATA ADR
004446' 004777  173424                  JSR     PC,@GETBYT              ;GET NEXT DATA BYTE IN R1
004452' 010067  174704                  MOV     R0,WRADR                ;STORE NEXT DATA BYTE ADR
004456' 005367  174702          10$:    DEC     WRBCNT                  ;DECR THE DATA BYTE COUNT
004462' 110164  000006                  MOVB    R1,TBUF(R4)             ;LOAD THE DATA BYTE
004466' 005267  174720          WINCBC: INC     BYWR+2                  ;ADD 1 TO TOTAL BYTES
004472' 005567  174712                  ADC     BYWR                    ;WRITTEN COUNT
004476' 005267  174666                  INC     WRSIZE                  ;ADD 1 TO THIS MSG'S SIZE
004502' 000663                          BR      INTEX                   ;GO TO INT EXIT

004504' 005767  174672          WCKPAD: TST     PADCNT                  ;ALREADY SENT A PAD CHAR?
004510' 001005                          BNE     WRRBSY                  ;N,Y-WRRBSY
004512' 005064  000006                  CLR     TBUF(R4)                ;LOAD A BYTE OF 0'S
004516' 005267  174660                  INC     PADCNT                  ;ADD 1 TO PAD CNT
004522' 000761                          BR      WINCBC                  ;GO INCR BYTE COUNTS
004524' 042767  000002  173250  WRRBSY: BIC     #WRBSY,DFLGWD           ;RESET WRITE BUSY FLAG
004532' 052767  000010  173242          BIS     #WRTERM,DFLGWD          ;SET DO WRITE TERMINATION FLAG
004540' 042764  000161  000004          BIC     #TSEND+TINTEN+TDNAIE+BRK,TCSR(R4) ;RESET SEND, WR INT ENB, & BRK
004546' 042714  000004                  BIC     #RQTS,(R4)              ;RESET REQ TO SEND
004552' 032714  000100                  BIT     #RINTEN,(R4)            ;READ INT ENABLE SET?
004556' 001633                          BEQ     CLRWTF                  ;N,Y-CLRWTF
004560' 032767  000001  173214          BIT     #RDBSY,DFLGWD           ;IS READ BUSY?
004566' 001627                          BEQ     CLRWTF                  ;Y,N-CLRWTF
004570' 000630                          BR      INTEX                   ;GO TO INT EXIT

004572' 005267  174636          DNAERR: INC     DNACNT                  ;ADD 1 TO DNA ERROR CNTR
004576' 052767  000100  173176          BIS     #WRIERR,DFLGWD          ;SET THE WRITE INT ERROR FLAG
004604' 000747                          BR      WRRBSY                  ;GO TERMINATE THE WRITE
```

```
                                        .SBTTL  SUBROUTINES FOR DU11 FUNCTION ROUTINES


                                        ;CHECK IF DU11 READ OR WRITE IS BUSY AND WAIT IF IT IS
                                                ;JSR    PC,CKRBSY       READ BUSY S/R CALL
                                                ;         ,CKWBSY       WRITE BUSY S/R CALL
                                                ;         ,CKBSY        BOTH BUSY S/R CALL
                                                ;
                                                ;DESTROYS R0,R3,R4
                                                ;
                                                ;ON EXIT:  R3 = PROG TBL ADR
                                                ;          R4 = RCSR ADR

004606' 012700  000001          CKRBSY: MOV     #1,R0                   ;SET UP CK READ FLAG
004612' 000405                          BR      CKCOM                   ;GO TO COMMON POINT

004614' 012700  100000          CKWBSY: MOV     #100000,R0              ;SET UP CK WRITE FLAG
004620' 000402                          BR      CKCOM                   ;GO TO COMMON POINT

004622' 012700  100001          CKBSY:  MOV     #100001,R0              ;SET UP CK RD & WR FLAGS

004626' 004767  001136          CKCOM:  JSR     PC,SUPTAD               ;SET UP PROG TBL & RCSR ADR'S
004632' 105700                  10$:    TSTB    R0                      ;CHECK FOR READ INT ENABLE?
004634' 001403                          BEQ     20$                     ;Y,N-20$
004636' 032714  000100                  BIT     #RINTEN,(R4)            ;READ INT ENABLE ON?
004642' 001006                          BNE     30$                     ;N,Y-30$
004644' 005700                  20$:    TST     R0                      ;CHECK FOR WRITE INT ENABLE?
004646' 100007                          BPL     40$                     ;Y,N-40$
004650' 032764  000140  000004          BIT     #TINTEN+TDNAIE,TCSR(R4) ;WRITE INT ENABLES SET?
004656' 001403                          BEQ     40$                     ;Y,N-40$
004660' 004577  173162          30$:    JSR     R5,@CIOBSY              ;RELEASE CONTROL
004664' 000762                          BR      10$                     ;GO CK AGAIN
004666' 032767  000014  173106  40$:    BIT     #RDTERM+WRTERM,DFLGWD   ;HAVE TO PROCESS PREV TERMINATION?
004674' 001405                          BEQ     STMADR                  ;Y,N-STMADR
004676' 010046                          MOV     R0,-(SP)                ;SAVE RD/WR CK FLAGS
004700' 004767  000020                  JSR     PC,PROCTM               ;GO PROCESS TERMINATION
004704' 012600                          MOV     (SP)+,R0                ;RESTORE RD/WR FLAGS
004706' 000751                          BR      10$                     ;GO RECHECK INT ENABLE
004710' 010567  174440          STMADR: MOV     R5,OBJADR               ;SAVE CURR USER STMNT ADR
004714' 162767  000004  174432          SUB     #4,OBJADR
004722' 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
                                        ;
004724' 010146                  PROCTM: MOV     R1,-(SP)                ;SAVE R1 & R2
004726' 010246                          MOV     R2,-(SP)
004730' 032767  000004  173044  10$:    BIT     #RDTERM,DFLGWD          ;NEED TO DO READ TERMINATION?
004736' 001433                          BEQ     20$                     ;Y,N-20$
004740' 042767  000004  173034          BIC     #RDTERM,DFLGWD          ;RESET READ'S PROC TERM FLAG
004746' 016767  174414  173044          MOV     RDSIZE,SIZE             ;STORE # OF BYTES READ
004754' 004767  000136                  JSR     PC,RRINTV               ;RESET READ'S INT VECTOR
004760' 032767  000020  173014          BIT     #RDIERR,DFLGWD          ;WAS THERE A READ INT ERROR?
004766' 001417                          BEQ     20$                     ;Y,N-20$
004770' 032767  000040  173004          BIT     #RDIDSC,DFLGWD          ;WAS IT A DATA SET CHG?
004776' 001404                          BEQ     14$                     ;Y,N-14$
005000' 004567  000274                  JSR     R5,ERRRIS               ;REPORT DSC ERROR ON READ
005004' 001436                          .WORD   DSCMSG-ERMBAS
005006' 000403                          BR      16$                     ;GO RESET ERROR FLGS
005010' 004567  000264          14$:    JSR     R5,ERRRIS               ;GO ISSUE READ XFER ERROR MSG WITH
005014' 001406                          .WORD   RXFERR-ERMBAS           ;READ'S INT STATUS
005016' 042767  000060  172756  16$:    BIC     #RDIERR+RDIDSC,DFLGWD   ;RESET READ INT ERR FLAGS
005024' 000426                          BR      30$                     ;GO TO EROR POINT
005026' 032767  000010  172746  20$:    BIT     #WRTERM,DFLGWD          ;NEED TO DO WRITE TERMINATION?
005034' 001425                          BEQ     PROCEX                  ;Y,N-PROCEX
005036' 042767  000010  172736          BIC     #WRTERM,DFLGWD          ;RESET WRITE'S PROC TERM FLAG
005044' 016767  174320  172746          MOV     WRSIZE,SIZE             ;STORE # OF BYTES WRITTEN
005052' 004767  000064                  JSR     PC,RWINTV               ;RESET WRITE'S INT VECTOR
005056' 032767  000100  172716          BIT     #WRIERR,DFLGWD          ;WAS THERE A WRITE INT ERROR?
005064' 001411                          BEQ     PROCEX                  ;Y,N-PROCEX
005066' 004567  000224                  JSR     R5,ERRWIS               ;GO ISSUE WRITE XFER ERROR MSG WITH
005072' 001467                          .WORD   WXFERR-ERMBAS           ;WRITE'S INT STATUS
005074' 042767  000100  172700          BIC     #WRIERR,DFLGWD          ;RESET WRITE INT ERR FLAG
005102' 004577  172742          30$:    JSR     R5,@CUPGER              ;GO TO MPG ERR RETN POINT
005106' 000710                          BR      10$                     ;GO CK FOR ANOTHER TERMINATION
005110' 012602                  PROCEX: MOV     (SP)+,R2                ;RESTORE R1 & R2
005112' 012601                          MOV     (SP)+,R1
005114' 000207                          RTS     PC                      ;EXIT IN-LINE
```

```
                                        ;RESET INTERRUPT VECTORS S/R'S
                                        ;
                                        ;JSR    PC,RRINTV       READ VECTOR S/R CALL
                                        ;         ,RWINTV       WRITE VECTOR S/R CALL
                                        ;
                                        ;R3 = PROG TBL ADR
                                        ;
                                        ;DESTROYS R0

005116' 004567 000052          RRINTV: JSR     R5,TRVECT               ;GO CK IF I HAVE VECTOR CONTROL
005122' 000406                         BR      20$                     ;BR IF I DON'T
005124' 016767 172676 000004           MOV     IVCTAD,10$              ;GET READ INT VECT ADR
005132' 004577 172732                  JSR     R5,@CLRVEC              ;GO HAVE MPG CLEAR IT
005136' 000000                 10$:    .WORD   XXXX
005140' 000207                 20$:    RTS     PC                      ;EXIT IN-LINE

005142' 004567 000056          RWINTV: JSR     R5,TWVECT               ;GO CK IF I HAVE VECTOR CONTROL
005146' 000411                         BR      40$                     ;BR IF I DON'T
005150' 016767 172652 000012           MOV     IVCTAD,30$              ;GET READ INT VECT ADR
005156' 062767 000004 000004           ADD     #4,30$                  ;POINT IT AT WRITE'S VECTOR
005164' 004577 172700                  JSR     R5,@CLRVEC              ;GO HAVE MPG CLEAR IT
005170' 000000                 30$:    .WORD   XXXX
005172' 000207                 40$:    RTS     PC                      ;EXIT IN-LINE


                                        ;TEST INTERRUPT VECTORS S/R'S
                                        ;
                                        ;JSR    R5,TRVECT       READ VECTOR S/R CALL
                                        ;         ,TWVECT       WRITE VECTOR S/R CALL
                                        ;BR     LABEL           EXECUTED IF NOT SAME
                                        ;
                                        ;R3 = PROG TBL ADR
                                        ;
                                        ;DESTROYS R0

005174' 016767 172626 000010   TRVECT: MOV     IVCTAD,10$              ;GET READ INT VECT ADR
005202' 016346 000004                  MOV     PFWADR(R3),-(SP)        ;STORE FLGWD ADR TO IDENTIFY ME
005206' 004577 172660                  JSR     R5,@TSTVEC              ;DO I HAVE VECTOR CONTROL?
005212' 000000                 10$:    .WORD   XXXX                    ;  MPG WILL TELL ME SINCE I CAN'T
005214' 176644                         .WORD   DURINT-.                ;  GET AT LOWER MEM IF MEM MGMNT
005216' 000401                         BR      20$                     ;BR IF I DONT'T HAVE CNTRL
005220' 005725                         TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
005222' 000205                 20$:    RTS     R5                      ;EXIT IN-LINE

005224' 016767 172576 000016   TWVECT: MOV     IVCTAD,30$              ;GET INT VECT BASE ADR
005232' 062767 000004 000010           ADD     #4,30$                  ;POINT IT AT WRITE'S VECTOR
005240' 016346 000004                  MOV     PFWADR(R3),-(SP)        ;STORE FLGWD ADR TO IDENTIFY ME
005244' 004577 172622                  JSR     R5,@TSTVEC              ;DO I HAVE VECTOR CONTROL?
005250' 000000                 30$:    .WORD   XXXX                    ;  MPG WILL TELL ME SINCE I CAN'T
005252' 177070                         .WORD   DUWINT-.                ;  GET AT LOWER MEM IF MEM MGMNT
005254' 000401                         BR      40$                     ;BR IF I DONT'T HAVE CNTRL
005256' 005725                         TST     (R5)+                   ;BYPASS BR INST IN S/R CALL
005260' 000205                 40$:    RTS     R5                      ;EXIT IN-LINE
```

```
1314
1315
1316                                        ;ERROR INFORMATION DISPLAY S/R
1317
1318                                        ;JSR    R5,ERRCS        S/R CALL FOR CURR STATUS
1319                                        ;         ,ERRRIS       S/R CALL FOR READ INT STATUS
1320                                        ;         ,ERRWIS       S/R CALL FOR WRITE INT STATUS
1321                                        ;.WORD  MSGADR-ERMBAS   REL ADR OF ERROR MSG
1322                                        ;
1323                                        ;R3 = PROG TABLE ADR
1324                                        ;
1325                                        ;DESTROYS R0,R1,R2
1326
1327  005262' 012767  173530  000326  ERRCS:  MOV     #CSTAT-ERSTAD,ERSTAD    ;STORE ADRS OF CURR STATUS
1328  005270' 012767  173674  000156          MOV     #CSTAT-EBSBAS,EBSTAT
1329  005276' 000415                          BR      ERRCOM                  ;GO TO COMMON POINT
1330
1331  005300' 012767  173514  000310  ERRRIS: MOV     #RISTAT-ERSTAD,ERSTAD   ;STORE ADRS OF LAST READ
1332  005306' 012767  173660  000140          MOV     #RISTAT-EBSBAS,EBSTAT   ;INT STATUS
1333  005314' 000406                          BR      ERRCOM                  ;GO TO COMMON POINT
1334
1335  005316' 012767  173522  000272  ERRWIS: MOV     #WISTAT-ERSTAD,ERSTAD   ;STORE ADRS OF LAST WRITE
1336  005324' 012767  173666  000122          MOV     #WISTAT-EBSBAS,EBSTAT   ;INT STATUS
1337
1338  005332' 012567  000062          ERRCOM: MOV     (R5)+,ERMBAS            ;STORE MSG ADR
1339  005336' 012767  000001  172456          MOV     #1,ERRI                 ;SET THE ERROR INDICATOR
1340  005344' 032763  020400  000002          BIT     #DOERCK+PRONER,POPSW(R3) ;ERROR CHECKING OR PRINTING INHIBITED?
1341  005352' 001152                          BNE     ERREX                   ;Y,N-ERREX
1342  005354' 010446                          MOV     R4,-(SP)                ;SAVE R4 & R5
1343  005356' 010546                          MOV     R5,-(SP)
1344  005360' 005004                          CLR     R4                      ;SET USER MODE PRINT FLAG
1345  005362' 004767  000442                  JSR     PC,DISUNM               ;DISPLAY DEVICE I.D.
1346  005366' 010700                          MOV     PC,R0                   ;GET START ADR OF ERROR MSG
1347  005370' 062700  000030                  ADD     #ERMBAS-.,R0
1348  005374' 061000                          ADD     (R0),R0
1349  005376' 012701  177777                  MOV     #-1,R1                  ;INITIALIZE MSG LENGTH
1350  005402' 005201                  10$:    INC     R1                      ;ADD 1 TO MSG LENGTH
1351  005404' 105720                          TSTB    (R0)+                   ;MSG TERMINATOR?
1352  005406' 001375                          BNE     10$                     ;Y,N-10$
1353  005410' 010167  000006                  MOV     R1,ERMBAS+2             ;STORE MSG LENGTH
1354  005414' 004567  000536                  JSR     R5,PRINT                ;PRINT ERROR MSG SPECIFIED
1355  005420' 000000                  ERMBAS: .WORD   XXXX
1356  005422' 000000                          .WORD   XXXX
1357  005424' 026727  177770  001577          CMP     ERMBAS,#IVMODE-ERMBAS   ;INV MODE MSG OR HIGHER?
1358  005432' 103072                          BHIS    ERRSNM                  ;N,Y-ERRSNM
1359  005434' 010701                          MOV     PC,R1                   ;GET ADR OF CODE  AREA IN ERR MSG
1360  005436' 062701  001535                  ADD     #CODFLD-.,R1
1361  005442' 010700                          MOV     PC,R0                   ;SET UP ADR OF ERROR CODE TBL
1362  005444' 062700  000236                  ADD     #ERCDTB-.,R0
1363  005450' 010702                          MOV     PC,R2                   ;SET UP ADR OF STORED DEV REG'S
1364  005452' 062702                  EBSBAS: ADD     (PC)+,R2
1365  005454' 173674                  EBSTAT: .WORD   CSTAT-EBSBAS
1366  005456' 012767  000013  000124          MOV     #11.,70$                ;INITIALIZE MSG LENGTH
1367  005464' 012746  000024                  MOV     #20.,-(SP)              ;INITIALIZE CODE FIELD CNT
1368  005470' 012205                  15$:    MOV     (R2)+,R5                ;GET NEXT DEV REG WORD
1369  005472' 000305                          SWAB    R5                      ;GET DESIRED BYTE IN LOW BYTE
```

```
1370  005474' 112004                20$:    MOVB    (R0)+,R4        ;GET FLAG & LENGTH BYTE
1371  005476' 005704                        TST     R4              ;END OF THE CODE TBL?
1372  005500' 001434                        BEQ     60$             ;N,Y-60$
1373  005502' 122704  000377                CMPB    #377,R4         ;GO TO NXT DEV REG WORD?
1374  005506' 001770                        BEQ     15$             ;N,Y-15$
1375  005510' 131005                        BITB    (R0),R5         ;THIS ERROR BIT SET IN DEV REG BYTE?
1376  005512' 001004                        BNE     40$             ;N,Y-40$
1377  005514' 042704  177770                BIC     #177770,R4      ;ISOLATE ENTRY LENGTH
1378  005520' 060400                        ADD     R4,R0           ;POINT AT NXT CODE TBL ENTRY
1379  005522' 000764                        BR      20$             ;GO CK FOR NXT CODE
1380  005524' 042704  177770        40$:    BIC     #177770,R4      ;ISOLATE I.D. NAME LENGTH + 1
1381  005530' 020416                        CMP     R4,(SP)         ;ENOUGH ROOM FOR NAME?
1382  005532' 101017                        BHI     60$             ;Y,N-60$
1383  005534' 060467  000050                ADD     R4,70$          ;ADJ MSG LENGTH FOR NAME
1384  005540' 005304                        DEC     R4              ;ADJ FOR BIT MASK CHAR
1385  005542' 005200                        INC     R0              ;POINT PAST BIT MASK
1386  005544' 021627  000024                CMP     (SP),#20.       ;FIRST ERROR CODE IN MSG?
1387  005550' 001403                        BEQ     50$             ;N,Y-50$
1388  005552' 112721  000054                MOVB    #',,(R1)+       ;MOVE COMMA TO MSG
1389  005556' 005316                        DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
1390  005560' 112021                50$:    MOVB    (R0)+,(R1)+     ;MOVE ERROR CODE TO MSG
1391  005562' 005316                        DEC     (SP)            ;ADJ REMAINING ROOM IN MSG
1392  005564' 005304                        DEC     R4              ;MOVED ALL NAME CHARS?
1393  005566' 001374                        BNE     50$             ;Y,N-50$
1394  005570' 000741                        BR      20$             ;GO CK FOR MORE ERROR BITS
1395  005572' 005004                60$:    CLR     R4              ;SET USER MODE PRINT
1396  005574' 022627  000024                CMP     (SP)+,#20.      ;ANY ERROR CODES PUT IN MSG?
1397  005600' 001404                        BEQ     80$             ;Y,N-80$
1398  005602' 004567  000350                JSR     R5,PRINT        ;GO ISSUE ERROR BITS MSG
1399  005606' 001351                        .WORD   ERRMSG-.
1400  005610' 000040                70$:    .WORD   32.
1401  005612' 004567  000236        80$:    JSR     R5,DISPST       ;DISPLAY DEVICE REG'S
1402  005616' 000000                ERRSTAD: .WORD  XXXX
1403  005620' 016300  000022        ERRSNM: MOV     PSRCST(R3),R0   ;GET ADR OF SRC STMNTS
1404  005624' 111001                110$:   MOVB    (R0),R1         ;SAVE STMNT LENGTH
1405  005626' 026067  000004  173520        CMP     4(R0),OBJADR    ;ERROR OCCUR ON THIS STMNT?
1406  005634' 001402                        BEQ     120$            ;N,Y-120$
1407  005636' 060100                        ADD     R1,R0           ;POINT AT NXT STMNT
1408  005640' 000771                        BR      110$            ;GO CK NXT STMNT
1409  005642' 005720                120$:   TST     (R0)+           ;SET UP ADR OF STMNT # DATA
1410  005644' 010701                        MOV     PC,R1           ;SET UP DATA OUTPUT ADR
1411  005646' 062701  001152                ADD     #STMNUM-.,R1
1412  005652' 004577  172204                JSR     R5,@DECASC      ;CONVERT IT TO ASCII
1413  005656' 012767  020040  001140        MOV     #20040,STMNUM+4 ;SET 2 LOW DIGITS TO SPACES
1414  005664' 004567  000266                JSR     R5,PRINT        ;ISSUE STMNT # MSG
1415  005670' 001120                        .WORD   STMNMG-.
1416  005672' 177762                        .WORD   -14.
1417  005674' 012605                        MOV     (SP)+,R5        ;RESTORE R5 & R4
1418  005676' 012604                        MOV     (SP)+,R4
1419  005700' 000205                ERREX:  RTS     R5              ;EXIT IN-LINE
```

```
1421
1422
1423                                    ;ERROR MESSAGE CODE TABLE
1424
1425                                            ;377 = GO TO NEXT DEVICE REGISTER WORD
1426                                            ;
1427                                            ;BYTE 0 CONTAINS FLAG BITS & I.D. NAME LENGTH
1428                                            ;
1429                                            ;       BITS 0-2 = LENGTH OF BIT MASK + I.D. NAME
1430                                            ;
1431                                            ;BYTE 1 IS THE BIT MASK
1432                                            ;
1433                                            ;BYTES 2 THRU ? ARE THE BIT'S ASCII I.D.
1434
1435  005702'    377                  ERCDTB: .BYTE   377
1436  005703'    003  051200    130          .ASCII  <003><200>/RX/          ;RBUF: BITS 15 - 8
1437  005707'    004  047500  051126         .ASCII  <004><100>/OVR/
1438  005714' 020004  051106    115          .ASCII  <004><040>/FRM/
1439  005721'    004  050020  051101         .ASCII  <004><020>/PAR/
1440  005726'    377                         .BYTE   377
1441  005727'    004  042200  040516         .ASCII  <004><200>/DNA/         ;TCSR: BITS 15 - 8
1442  005734'    000                         .BYTE   0                       ;TABLE TERMINATOR
1443          005736'                       .EVEN
```

```
                                        .SBTTL  SUBROUTINES FOR DU11 DEVICE ROUTINE


                                        ;SAVE REGISTERS R0 THRU R5
                                                ;JSR    R0,SAVREG       S/R CALL
005736' 010146                  SAVREG: MOV     R1,-(SP)                ;SAVE R0 THRU R5
005740' 010246                          MOV     R2,-(SP)
005742' 010346                          MOV     R3,-(SP)
005744' 010446                          MOV     R4,-(SP)
005746' 010546                          MOV     R5,-(SP)
005750' 010007                          MOV     R0,PC                   ;EXIT IN-LINE


                                        ;RESTORE REGISTERS R0 THRU R5
                                                ;JSR    R0,RESREG       S/R CALL
005752' 005726                  RESREG: TST     (SP)+                   ;RESTORE R5 THRU R0
005754' 012605                          MOV     (SP)+,R5
005756' 012604                          MOV     (SP)+,R4
005760' 012603                          MOV     (SP)+,R3
005762' 012602                          MOV     (SP)+,R2
005764' 012601                          MOV     (SP)+,R1
005766' 000200                          RTS     R0                      ;EXIT IN-LINE


                                        ;SET PROGRAM'S PROG TABLE ADR IN R3 & RCSR ADR IN R4
                                                ;JSR    PC,SUPTAD       S/R CALL
005770' 010703                  SUPTAD: MOV     PC,R3                   ;SET UP LOCATION ZERO ADR
005772' 062703  172006                  ADD     #LOCZ-.,R3
005776' 166303  177776                  SUB     -2(R3),R3               ;SUBTRACT PROG TBL LENGTH
006002' 016704  172016                  MOV     DREGAD,R4               ;GET DEV REG BASE ADR (RCSR)
006006' 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
006010' 010501                  STSTAT: MOV     R5,R1                   ;GET REL STORAGE ADR & MAKE
006012' 062501                          ADD     (R5)+,R1                ;IT ABSOLUTE
006014' 016700  172004                  MOV     DREGAD,R0               ;GET DEV REG ADR
006020' 012021                          MOV     (R0)+,(R1)+             ;STORE ALL READABLE DEV REG'S
006022' 012021                          MOV     (R0)+,(R1)+
006024' 012021                          MOV     (R0)+,(R1)+
006026' 000205                          RTS     R5                      ;EXIT IN-LINE
```

```
1500
1501
1502                                    ;DISPLAY DEVICE I.D. AND DEVICE REGISTER ADDRESS
1503
1504                                            ;JSR    PC,DISUNM       S/R CALL
1505                                            ;
1506                                            ;R4 = CMND/USER MODE PRINT FLAG
1507                                            ;R3 = PROG TBL ADR
1508                                            ;
1509                                            ;DESTROYS R0,R1,R2
1510
1511  006030' 016700  171770        DISUNM: MOV     DREGAD,R0               ;GET DU11 DEV REG ADR
1512  006034' 004577  172016                JSR     R5,@BINASC              ;CONVERT BINARY # TO ASCII
1513  006040' 000345                        .WORD   UNASCI-.
1514  006042' 004567  000110                JSR     R5,PRINT                ;GO ISSUE DEV I.D. MSG
1515  006046' 000323                        .WORD   UNITMG-.
1516  006050' 000022                        .WORD   18.
1517  006052' 000207                        RTS     PC                      ;EXIT IN-LINE
1518
1519
1520                                    ;TAILOR STATUS MSG & PRINT IT
1521
1522                                            ;JSR    R5,DISPST       S/R CALL
1523                                            ;.WORD  STATADR-.       REL ADR OF STATUS DATA
1524                                            ;
1525                                            ;DESTROYS R0,R1,R2
1526
1527  006054' 010502                DISPST: MOV     R5,R2                   ;GET REL DATA ADR
1528  006056' 062502                        ADD     (R5)+,R2                ;MAKE IT ABS
1529  006060' 010701                        MOV     PC,R1                   ;SET UP ADR OF REG NAMES IN ASCII
1530  006062' 062701  172034                ADD     #DVREGS-.,R1
1531  006066' 012746  000003                MOV     #3,-(SP)                ;STORE # OF REGISTERS TO DISPLAY
1532  006072' 012167  000316        10$:    MOV     (R1)+,DVRGMG            ;MOVE REG NAME TO MSG
1533  006076' 012167  000314                MOV     (R1)+,DVRGMG+2
1534  006102' 005721                        TST     (R1)+                   ;BYPASS DISP VALUE
1535  006104' 012200                        MOV     (R2)+,R0                ;GET REG'S STORED VALUE
1536  006106' 010746                        MOV     PC,-(SP)                ;SET UP ADR OF WRITE
1537  006110' 062716  173234                ADD     #WITCSR-.,(SP)          ;INT STATUS
1538  006114' 022602                        CMP     (SP)+,R2                ;THIS THE UNUSED WD IN WR INT?
1539  006116' 001413                        BEQ     20$                     ;N,Y-20$
1540  006120' 010146                        MOV     R1,-(SP)                ;SAVE R1 & R2
1541  006122' 010246                        MOV     R2,-(SP)
1542  006124' 004577  171726                JSR     R5,@BINASC              ;CONVERT IT TO ASCII
1543  006130' 000272                        .WORD   DVRGDT-.
1544  006132' 004567  000020                JSR     R5,PRINT                ;PRINT THE STATUS MSG
1545  006136' 000256                        .WORD   DVRGMG-.
1546  006140' 000014                        .WORD   12.
1547  006142' 012602                        MOV     (SP)+,R2                ;RESTORE R1 & R2
1548  006144' 012601                        MOV     (SP)+,R1
1549  006146' 005316                20$:    DEC     (SP)                    ;DECR REG CNT
1550  006150' 001350                        BNE     10$                     ;DONE ALL? (Y,N-10$)
1551  006152' 005726                        TST     (SP)+                   ;REMOVE CNT FROM STACK
1552  006154' 000205                        RTS     R5                      ;EXIT IN-LINE
```

```
1554
1555
1556                                    ;ISSUE MSG TO LIST DEVICE SUBROUTINE
1557
1558                                    ;JSR    R5,PRINT        S/R CALL
1559                                    ;.WORD  MSGADR-.        REL ADR OF MSG
1560                                    ;.WORD  BYTCNT          MSG BYTE CNT (IF NEGATIVE,
1561                                    ;                         RESET PRT DEV DEDICATED.)
1562                                    ;
1563                                    ;R3 = PROG TBL ADR
1564                                    ;R4 = FLAGWORD -- IF NEGATIVE, USE CMND MODE PRINT
1565                                    ;
1566                                    ;DESTROYS R0,R1,R2
1567
1568  006156' 010500          PRINT:    MOV     R5,R0                   ;GET MSG ADR & MAKE IT ABS
1569  006160' 062500                    ADD     (R5)+,R0
1570  006162' 012501                    MOV     (R5)+,R1                ;GET BYTE COUNT
1571  006164' 005704                    TST     R4                      ;USE CMND MODE PRINT?
1572  006166' 100030                    BPL     40$                     ;Y,N-40$
1573  006170' 010702                    MOV     PC,R2                   ;SET UP LINK INFO ADR
1574  006172' 062702  000040            ADD     #20$-.,R2
1575  006176' 160200                    SUB     R2,R0                   ;MAKE MSG ADR REL
1576  006200' 010022                    MOV     R0,(R2)+                ;STORE MSG ADR
1577  006202' 010112                    MOV     R1,(R2)                 ;STORE MSG'S BYTE COUNT
1578  006204' 100001                    BPL     10$                     ;CNT NEG? (Y,N-10$)
1579  006206' 005412                    NEG     (R2)                    ;MAKE IT POSITIVE
1580  006210' 016367  000006  000056 10$:  MOV   PASCIN(R3),PROGNM      ;STORE PROG'S # IN MSG
1581  006216' 004577  171632            JSR     R5,@CLIST               ;ISSUE PROG #
1582  006222' 000050                    .WORD   PNMMSG-.
1583  006224' 000005                    .WORD   5
1584  006226' 004577  171622            JSR     R5,@CLIST               ;ISSUE MSG SPECIFIED
1585  006232' 000000          20$:      .WORD   XXXX
1586  006234' 000000                    .WORD   XXXX
1587  006236' 004577  171612            JSR     R5,@CLIST               ;ISSUE A <CR> & <LF>
1588  006242' 000240                    .WORD   CRLF-.
1589  006244' 000002                    .WORD   2
1590  006246' 000410                    BR      PRTEX                   ;GO TO EXIT
1591  006250' 010067  000010  40$:      MOV     R0,50$                  ;STORE MSG'S ABS ADR
1592  006254' 010167  000006            MOV     R1,60$                  ;STORE ITS BYTE CNT
1593  006260' 004577  171566            JSR     R5,@ULIST               ;GO TO MPG TO ISSUE THE MSG
1594  006264' 000000          50$:      .WORD   XXXX
1595  006266' 000000          60$:      .WORD   XXXX
1596  006270' 000205          PRTEX:    RTS     R5                      ;EXIT IN-LINE
```

```
                                                .SBTTL  DU11 MESSAGE STORAGE AREA
                                                .NLIST  BEX
                                                .EVEN
006272' 021520                          PNMMSG: .ASCII  /P#/
006274' 054130      011                 PROGNM: .ASCII  /XX/<011>
006277'    101  020124  040514          LRIMSG: .ASCII  /AT LAST READ INT:/
006320' 052101  046040  051501          LWIMSG: .ASCII  /AT LAST WRITE INT:/
006342' 052503  051122  047105          CURMSG: .ASCII  /CURRENTLY:/
006354' 047105  020104  043117          RENDMG: .ASCII  /END OF REPORT/
006371'    052  025052  042052          UNITMG: .ASCII  /****DU11 AT /
006405'    130  054130  054130          UNASCI: .ASCII  /XXXXXX/
        006414'                                 .EVEN
006414' 054130  054130  020075          DVRGMG: .ASCII  /XXXX= /
006422' 054130  054130  054130          DVRGDT: .ASCII  /XXXXXX/
006430' 054502  042524  035123          CNTSMG: .ASCII  /BYTES:  RD= /
006444' 054130  054130  054130          BCMRD:  .ASCII  /XXXXXXXXXXXX  WR= /
006466' 054130  054130  054130          BCMWR:  .ASCII  /XXXXXXXXXXXX/
006502' 005015                          CRLF:   .ASCII  <015><012>
006504' 041411  047115  051504                  .ASCII  <011>/CMNDS:  RD= /
006521'    130  054130  054130          CMDCRD: .ASCII  /XXXXXX WR= /
006534' 054130  054130  054130          CMDCWR: .ASCII  /XXXXXX BRK= /
006550' 054130  054130  054130          CMDBRK: .ASCII  /XXXXXX MISC= /
006565'    130  054130  054130          CMDCMS: .ASCII  /XXXXXX/<015><012>
006575'    011  051105  047522                  .ASCII  <011>/ERRORS: PAR= /
006613'    130  054130  054130          CNTPAR: .ASCII  /XXXXXX  FRM= /
006630' 054130  054130  054130          CNTFRM: .ASCII  /XXXXXX  OVR= /
006645'    130  054130  054130          CNTOVR: .ASCII  /XXXXXX  DSC= /
006662' 054130  054130  054130          CNTDSC: .ASCII  /XXXXXX/<015><012><011><011>/DNA= /
006701'    130  054130  054130          CNTDNA: .ASCII  'XXXXXX  T/O= '
006716' 054130  054130  054130          CNTTOE: .ASCII  /XXXXXX  DATA= /
006734' 054130  054130  054130          CNTDER: .ASCII  /XXXXXX/<015><012>
006744' 044411  052116  051105                  .ASCII  <011>/INTERRUPTS:  RD= /
006766' 054130  054130  054130          CNTRDI: .ASCII  /XXXXXX  WR= /
007002' 054130  054130  054130          CNTWRI: .ASCII  /XXXXXX/
        007010'                         CNTSEN= .
                                                .EVEN
007010' 052123  047115  020124          STMNMG: .ASCII  /STMNT # /
007020' 054130  054130  054130          STMNUM: .ASCII  /XXXXXX/
007026' 051105  047522  020122          RXFERR: .ASCIZ  'ERROR ON READ DATA XFER'
007056' 040504  040524  051440          DSCMSG: .ASCIZ  'DATA SET CHG INT ON READ'
007107'    105  051122  051117          WXFERR: .ASCIZ  'ERROR ON WRITE DATA XFER'
007140' 044524  042515  052517          IOTO:   .ASCIZ  'TIMEOUT ON I/O'
007157'    105  051122  051117          ERBMSG: .ASCII  /ERROR BITS: /
007173' 000024                          CODFLD: .BLKB   20.
007217'    115  042117  020105          IVMODE: .ASCIZ  /MODE NOT 0, 2 OR 3/
007242' 020043  043117  041440          IVBITS: .ASCIZ  /# OF CHAR BITS NOT 5, 6, 7 OR 8/

                                                .EVEN

                                                .LIST   BEX


        007302'                         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

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

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

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

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

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

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

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

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

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

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

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

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

```
1710    000032                  PMDLCD= +26.    ;DEV ROUT MODEL # CODE - 1 WORD
1711
1712    000034                  PDPNTR= +28.    ;CURRENT DEVICE NUMBER POINTER - 1 BYTE
1713
1714    000035                  PCURDV= +29.    ;CURRENT DEVICE # - 1 BYTE
1715
1716    000036                  PDNUMS= +30.    ;DEVICE NUMBERS - 16 BYTES
1717
1718    000056                  PTEM0=  +46.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1719
1720    000060                  PTEM1=  +48.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1721
1722    000062                  PTEM2=  +50.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1723
1724    000064                  PTEM3=  +52.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1725
1726    000066                  PTEM4=  +54.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1727
1728    000070                  PTEM5=  +56.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1729
1730    000072                  PTEM6=  +58.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1731
1732    000074                  PTEM7=  +60.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1733
1734    000076                  PTEM8=  +62.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1735
1736    000100                  PTEM9=  +64.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1737
1738    000102                  PTEM10= +66.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1739
1740    000104                  PTEM11= +68.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1741
1742    000106                  PTEM12= +70.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1743
1744    000110                  PTEM13= +72.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1745
1746    000112                  PTEM14= +74.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1747
1748    000114                  PTEM15= +76.    ;USER PROGRAM TEMPORARY STORAGE - 1 WORD
1749
1750    000116                  PNBR=   +78.    ;NUMBER OF BYTES TO TRANSFER ON MOVE (NBR) - 1 WORD
1751
1752    000120                  PSRC=   +80.    ;DATA SOURCE ADDRESS ON MOVE (SRC) - 1 WORD
1753
1754    000122                  PDST=   +82.    ;DATA DESTINATION ADDRESS ON MOVE (DST) - 1 WORD
1755
1756    000124                  PSTKCT= +84.    ;# OF WORDS (X 2) SAVED OFF STACK - 1 WORD
1757
1758    000126                  PSTKSV= +86.    ;STACK WORDS STORAGE AREA - 30 WORDS
1759
1760    000222                  PSVREG= +146.   ;USER'S R0 THRU R5 REGISTERS STORAGE AREA - 6 WORDS
1761
1762    000236                  PUSRPC= +158.   ;USER'S CURRENT PROGRAM COUNTER - 1 WORD
1763
```

```
1765                                    ;FOLLOWING ENTRIES (PRDIOX THRU PUBMAP) ARE ONLY IN MEM MGMNT VERSION
1766
1767                                ;(PRDIOX= +160. ;18/22 BIT ABSOLUTE ADDRESS OF READ I/O AREA - 2 WORDS)
1768
1769                                ;(PRDIOV= +164. ;18 BIT VIRTUAL ADDRESS OF READ I/O AREA - 2 WORDS)
1770
1771                                ;(PWRIOX= +168. ;18/22 BIT ABSOLUTE ADDRESS OF WRITE I/O AREA - 2 WORDS)
1772
1773                                ;(PWRIOV= +172. ;18 BIT VIRTUAL ADDRESS OF WRITE I/O AREA - 2 WORDS)
1774
1775                                ;(PUPARS= +176. ;STORAGE AREA FOR USER'S PAR'S 0 THRU 7 - 8 WORDS)
1776
1777                                ;(PUPDRS= +192. ;STORAGE AREA FOR USER'S PDR'S 0 THRU 7 - 8 WORDS)
1778
1779                                ;(PUBMAP= +208. ;1ST UNIBUS MAP REG # AND # OF REGS USED - 1 WORD)
1780
1781                                    ;END OF MEM MGMNT ONLY ENTRIES
1782
1783        000240                  PTSIZE= +160.   ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - NON MEM MGMNT
1784
1785                                ;(PTSIZE= +210. ;PROGRAM TABLE SIZE IN BYTES - 1 WORD - MEM MGMNT VERSION)
1786
1787        000242                  PTEND=  +162.   ;END OF PROGRAM TABLE - NON MEM MGMNT VERSION
1788
1789                                ;(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
```

```
1847   000062                DVPDTA= +50.    ;CONVERT PACKED DECIMAL TO ASCII BR ADR (DECASC) - 1 WORD
1848
1849   000064                DVSFWD= +52.    ;MPG SYSTEM FLAGWORD ADDRESS (CSYSFW) - 1 WORD
1850
1851   000066                DVSVEC= +54.    ;SET INTERRUPT VECTOR BR ADR (SETVEC) - 1 WORD
1852
1853   000070                DVCVEC= +56.    ;CLEAR INTERRUPT VECTOR BR ADR (CLRVEC) - 1 WORD
1854
1855   000072                DVTVEC= +58.    ;TEST INTERRUPT VECTOR BR ADR (TSTVEC) - 1 WORD
1856
1857   000074                DVRINT= +60.    ;RETURN FROM INTERRUPT BR ADR (RTNINT) - 1 WORD
1858
1859   000076                DVGETB= +62.    ;GET DATA BYTE BR ADR (GETBYT) - 1 WORD
1860
1861   000100                DVPUTB= +64.    ;PUT DATA BYTE BR ADR (PUTBYT) - 1 WORD
1862
1863   000102                DEVSTP= +66.    ;DEVICE ROUT REL SYMBOL TABLE POINTER - 1 WORD
1864
1865   000104                DEVETP= +68.    ;DEVICE ROUT REL ENTRY TABLE POINTER - 1 WORD
1866
1867   000106                DVPTEP= +70.    ;PACK TABLE EXTEN. REL POINTER - 1 WORD
1868
1869   000110                DVVTEP= +72.    ;VECTOR TABLE EXTEN. REL POINTER - 1 WORD
1870
1871   000112                DVCTEP= +74.    ;COMPILER TBL EXTEN. REL POINTER - 1 WORD
1872
1873   000114                DVIWSP= +76.    ;DEVICE INTERFACE WORD SYMBOL TBL REL POINTER - 1 WORD
1874
1875   000116                DRTEND= +78.    ;END OF DEVICE ROUTINE TABLE
1876
1877
1878   000001                        .END
```

DTDUAA.P11 SYMBOL TABLE

```
ACTIVE= 100000          COUNTS  001404R 002 DSCMSG  007056R 002 GTNXTD= 001000          PATCH   001450R 002
ANSWER  002574R 002     CRESET  002770R 002 DSR   = 001000          HANGUP  002716R 002 PC    =%000007
AURPEP= 000010          CRLF    006502R 002 DTOEAD= 000044          HDUPLX  003050R 002 PCSR  = 000002
AUTORP= 000020          CSTAT   001346R 002 DTR   = 000002          HLFDPX= 000010          PCSRV   001446R 002
BCMRD   006444R 002     CSYSFW  000064R 002 DURINT  004060R 002 HSKEEP  001520R 002 PCURDV= 000035
BCMWR   006466R 002     CTPRIO= 000020      DUWINT  004342R 002 HSKPEN= 001446R 002 PDNUMS= 000036
BINASC  000056R 002     CTS   = 020000      DVBTDA= 000060      HSKPEP= 000004      PDPNTR= 000034
BITERR  003350R 002     CUPGER  000050R 002 DVCMDS  000154R 002 HSKPST= 001332R 002 PDST  = 000122
BITS    003266R 002     CURMSG  006342R 002 DVCPRT= 000054      ICONS   001556R 002 PFBBOV= 000002
BITS5 = 000000          CVSYNC  002426R 002 DVCPTE  001032R 002 INTEX   004252R 002 PFLGWD= 000000
BITS6 = 002000          CYCDVL= 002000      DVCTEP= 000112      IQTO    007140R 002 PFWADR= 000004
BITS7 = 004000          CYCPRG= 040000      DVCVEC= 000070      ISCNT   001400R 002 PLNGTH= 000026
BITS8 = 006000          DATAER  001440R 002 DVGETB= 000076      ISOCHM= 000000      PMDLCD= 000032
BITTBL  003366R 002     DECASC  000062R 002 DVIWSP= 000114      ISYNCM= 030000      PNAME = 000010
BIT11 = 004000          DECTAD= 000042      DVIWST  001306R 002 IVBITS  007242R 002 PNBR  = 000116
BIT12 = 010000          DERPAD= 000036      DVMVTE  000662R 002 IVCTAD  000026R 002 PNMMSG  006272R 002
BIT6  = 000100          DEVBTA= 000056      DVPDTA= 000062      IVMODE  007217R 002 POBJST= 000024
BREAK   003636R 002     DEVDER= 000050      DVPKTE  000342R 002 KILL    002262R 002 POPSW = 000002
BRK   = 000001          DEVDRA= 000024      DVPTEP= 000106      LDPCSR  003150R 002 PRDIOA= 000016
BRKCNT  001420R 002     DEVETP= 000104      DVPUTB= 000100      LISTEN  002626R 002 PRESET  003416R 002
BRKFLG= 000200          DEVFWD= 000002      DVREGE= 000154R 002 LOCZ    000000R 002 PRINT   006156R 002
BTASLZ  000060R 002     DEVIOB= 000046      DVREGS  000116R 002 LRIMSG  006277R 002 PROCEX  005110R 002
BYRD    001404R 002     DEVIVA= 000026      DVREND= 007302R 002 LWIMSG  006320R 002 PROCTM  004724R 002
BYWR    001410R 002     DEVIW1= 000004      DVREX   002120R 002 MAINTM= 014000      PROGNM  006274R 002
CACHE = 000200          DEVIW2= 000006      DVRGDT  006422R 002 MISCNT  001422R 002 PRONER= 020000
CALL    002574R 002     DEVIW3= 000010      DVRGMG  006414R 002 MODCOM  003242R 002 PRTEX   006270R 002
CARIER= 010000          DEVIW4= 000012      DVRINT= 000074      MODE    003202R 002 PSRC  = 000120
CIOBSY  000046R 002     DEVIW5= 000014      DVSFWD= 000064      MODEBT= 030000      PSRCST= 000022
CKBSY   004622R 002     DEVIW6= 000016      DVSVEC= 000066      MODERR  003256R 002 PSTKCT= 000124
CKCOM   004626R 002     DEVIW7= 000020      DVTVEC= 000072      MSFMT1  001330R 002 PSTKSV= 000126
CKRBSY  004606R 002     DEVIWB= 000022      DVUPRT= 000052      MSFMT2  001327R 002 PSVREG= 000222
CKWBSY  004614R 002     DEVRPS= 000030      DVVTEP= 000110      MSFMT3  001324R 002 PTEM0 = 000056
CLIST   000054R 002     DEVRSZ= 000000      EBSBAS  005452R 002 MSTRST= 000400      PTEM1 = 000060
CLRVEC  000070R 002     DEVSTP= 000102      EBSTAT  005454R 002 NIEREX  003356R 002 PTEM10= 000102
CLRWTF  004246R 002     DEVWPS= 000032      ERBMSG  007157R 002 NINTEX  002620R 002 PTEM11= 000104
CMASK   001374R 002     DFLGWD  000002R 002 ERCDTB  005702R 002 NOCOMP= 000001      PTEM12= 000106
CMDBRK  006550R 002     DHKPAD= 000034      ERMBAS  005420R 002 NOPAR   003142R 002 PTEM13= 000110
CMDCMS  006565R 002     DISCNT  002020R 002 ERRCOM  005332R 002 NORMAL  003064R 002 PTEM14= 000112
CMDCRD  006521R 002     DISPST  006054R 002 ERRCS   005262R 002 NOWAIT  002416R 002 PTEM15= 000114
CMDCWR  006534R 002     DISUNM  006030R 002 ERREX   005700R 002 NSTRIP  003024R 002 PTEM2 = 000062
CNTDER  006734R 002     DKILAD= 000040      ERRI    000022R 002 NSYNC = 000005      PTEM3 = 000064
CNTDMA  006701R 002     DMA   = 100000      ERRRIS  005300R 002 OBJADR  001354R 002 PTEM4 = 000066
CNTDSC  006662R 002     DNACNT  001434R 002 ERRSNM  005620R 002 OCPRES= 000100      PTEM5 = 000070
CNTFRM  006630R 002     DNAERR  004572R 002 ERRWIS  005316R 002 ODD    003124R 002 PTEM6 = 000072
CNTNUM= 000021          DOERCK= 000400      ERSTAD  005616R 002 OVR   = 040000      PTEM7 = 000074
CNTOVR  006645R 002     DOIOT = 000040      ERSTOP= 000004      OVRCNT  001430R 002 PTEM8 = 000076
CNTPAR  006613R 002     DREGAD  000024R 002 ESYNCM= 020000      PADCNT  001402R 002 PTEM9 = 000100
CNTRDI  006766R 002     DRTEND= 000116      EVEN    003114R 002 PAR   = 010000      PTEND = 000242
CNTSEN= 007010R 002     DRTLTH= 000116      FDUPLX  003034R 002 PARB    001376R 002 PTLGTH= 000242
CNTSMG  006430R 002     DSC   = 100000      FRM   = 020000      PARCNT  001424R 002 PTOCNT= 000030
CNTTOE  006716R 002     DSCCNT  001432R 002 FRMCNT  001426R 002 PARENB= 001000      PTSIZE= 000240
CNTWRI  007002R 002     DSCERR  004262R 002 GENPAR  002466R 002 PARSEN= 000400      PUSRPC= 000236
CODFLD  007173R 002     DSCIE = 000040      GETBYT  000076R 002 PASCIN= 000006      PUTBYT  000100R 002
```

DTDUAA.P11 SYMBOL TABLE

```
PWRIOA= 000020          REPORT  001614R  002  R4     =%000004          TBUF   = 000006          WISTAT  001340R  002
RBUF   = 000002          REPTBL  002130R  002  R5     =%000005          TCSR   = 000004          WITCSR  001344R  002
RCSR   = 000000          RESREG  005752R  002  SAVREG  005736R  002  TDNAIE= 000040          WRADR   001362R  002
RDADR   001356R  002    RING   = 040000       SCHSYC= 000020          TINTEN= 000100          WRBCNT  001364R  002
RDBCNT  001360R  002    RINTEN= 000100        SCNT    000006R  002  TOCNT   001372R  002  WRBRCM  003654R  002
RDBSY = 000001          RIRBUF  001334R  002  SEND    002672R  002  TOECNT  001436R  002  WRBSY = 000002
RDCNT   001414R  002    RIRCSR  001332R  002  SETDED= 000040          TOUTER  002172R  002  WRCNT   001416R  002
RDERR   004274R  002    RISTAT  001332R  002  SETVEC  000066R  002  TRVECT  005174R  002  WRICNT  001444R  002
RDE1    004314R  002    RITCSR  001336R  002  SIZE    000020R  002  TSEND = 000020          WRIERR= 000100
RDE2    004326R  002    RPTBAS  002064R  002  SP     =%000006          TSTVEC  000072R  002  WRITE   003616R  002
RDICNT  001442R  002    RPTEND  002110R  002  SPOPER= 000200          TWVECT  005224R  002  WRPSWD  000032R  002
RDIDSC= 000040          RPTLP   002046R  002  STMADR  004710R  002  ULIST   000052R  002  WRRBSY  004524R  002
RDIERR= 000020          RQTS   = 000004       STMNMG  007010R  002  UNASCI  006405R  002  WRSIZE  001370R  002
RDPSWD  000030R  002    RRINTV  005116R  002  STMNUM  007020R  002  UNITMG  006371R  002  WRTERM= 000010
RDRBSY  004206R  002    RTNINT  000074R  002  STONER= 100000          URSTOP= 000002          WT4IOT= 000010
RDSIZE  001366R  002    RWINTV  005142R  002  STRIP   003014R  002  USEUBM= 000200          WXDATA  004420R  002
RDTERM= 000004          RXER   = 100000       STRSYC= 000400          WAIT    002352R  002  WXFERR  007107R  002
RDWREX  003570R  002    RXFERR  007026R  002  STSTAT  006010R  002  WAITMD= 100000          XXXX   = 000000
READ    003446R  002    R0     =%000000       SUPTAD  005770R  002  WCKPAD  004504R  002  .      = 007302R  002

. ABS.  000000  000
        000000  001
DU11    007302  002


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

*,DTDUAA/NL:TOC/DOC=DTDUAA.P11
RUN-TIME:  5 11  1 SECONDS
RUN-TIME RATIO: 29/18=1.5
CORE USED:  5K  (9 PAGES)

DOCUMENT PAGES:  42
```

Spooler runtime 6 Seconds, 30 KCS, 172 disk reads, 3 disk writes, 42 pages