```
MTCTL  MAG. TAPE CONTROLLER    RT-11 MMAC  VM02-10   14-OCT-76 01:36:21 PAGE 28+
SYMBOL TABLE

NEWBFR= XXXXXX G      NFR8     0208R      NLPTR = XXXXXX G      NOKEY =  0080      NOOUT = XXXXXX G
NOWRIT=  0001         NTAPTR=  0008       NTATTR=  0004         NTDIMS=  0009       NTDLEN=  0005
NTLINK=  0000         NTNAME=  0002       NTPTR = XXXXXX G      NTRELY=  0010       NTSPTR=  0008
NTVAL =  0005         NTWCOL=  0007       NTWLEN=  0007         NTWROW=  0006       NULCHR= XXXXXX G
NULLTG= XXXXXX G      OBJBTR=  0002       OBJBCK=  0003         OBJDT =  0005       OBJLEN=  0000
OLD   = XXXXXX G      OLDCOD= XXXXXX G    OLDONE   06CCRG       OLDX     0768R      OLDXX    0760R
OLDXXX   075DR        ONSFLG=  0002       OPRADR= XXXXXX G      OUTBFR= XXXXXX G    PAETG = XXXXXX G
PBRM  =  0008         PGMBTR=  0002       PGMBP =  0005         PGMCD =  0009       PGMEVL= XXXXXX G
PGMFP =  0003         PGMLEN=  0000       PGMLNN=  0007         PGMPTR= XXXXXX G    PGMTG = XXXXXX G
PIAHX = XXXXXX G      PIAHY = XXXXXX G    PIALT = XXXXXX G      PIALX = XXXXXX G    PIALY = XXXXXX G
PIAMTA= XXXXXX G      PIAMTB= XXXXXX G    PLOSIG= XXXXXX G      PNDEOF= XXXXXX G    PNDFLG= XXXXXX G
PNTNTG= XXXXXX G      PNTSTG= XXXXXX G    POINT = XXXXXX G      PPMODE= XXXXXX G    PRGBLN   007ERG
PRGDAT   G092RG       PRGPGM   0088RG     PRIDEF=  G001         PRTTG = XXXXXX G    PSCTG = XXXXXX G
PSHFPN= XXXXXX G      PULFPN= XXXXXX G    PUPTAP= XXXXXX G      PUTBYT= XXXXXX G    RDBCRE   05E2R
RDBEND   05E7R        RDBEN.   05E6R      RDBFRV   05A5R        RDEOF    05EBR      READRS= XXXXXX G
RECCNT= XXXXXX G      RECLFG=  0004       REWIND= XXXXXX G      RFAIL =  00C6       RMAT  =  0040
RPTCTL=  0080         RSTR  =  0080       RTRNIG= XXXXXX G      RUNFLG=  0080       RUNN  =  0002
RWNDIT   0231R        R0    = XXXXXX G    R1    = XXXXXX G      R10   = XXXXXX G    R11   = XXXXXX G
R12   = XXXXXX G      R13   = XXXXXX G    R14   = XXXXXX G      R15   = XXXXXX G    R16   = XXXXXX G
R17   = XXXXXX G      R18   = XXXXXX G    R19   = XXXXXX G      R2    = XXXXXX G    R20   = XXXXXX G
R21   = XXXXXX G      R22   = XXXXXX G    R23   = XXXXXX G      R3    = XXXXXX G    R4    = XXXXXX G
R5    = XXXXXX G      R6    = XXXXXX G    R7    = XXXXXX G      R8    = XXXXXX G    R9    = XXXXXX G
SAVFNC=  0001         SBP   = XXXXXX G    SCOLER=  0040         SCRMS = XXXXXX G    SECDEF=  0002
SEMITG= XXXXXX G      SERCHS= XXXXXX G    SETARG= XXXXXX G      SETBIT   0612R      SNDBFR= XXXXXX G
SNDIT =  0080         SPCIOB= XXXXXX G    SPCIOF= XXXXXX G      STAT37= XXXXXX G    STKBLD= XXXXXX G
STPFLG=  0040         STPKEY=  0020       STRING=  0010         SYSERR= XXXXXX G    TABPTR= XXXXXX G
TAGPTR= XXXXXX G      TAPE     0000RG     TAPFIL   0768RG       TAPF2    076FRG     TCOL  = XXXXXX G
TLOON    0886R        TLIST    0847RG     TLISTT   085AR        TODCON   0733R      TODSEC   072FR
TOLD     0733R        TRANSI= XXXXXX G    TRCFLG=  0020         TST8NE= XXXXXX G    TYPARG= XXXXXX G
TYPASC   006ERG       TYPBIN   0066RG     TYPEXT   0787R        TYPLST   0076RG     TYPNEW   005ERG
TYPOLD   071CR        TYPSEC   009CRG     TSTDEV=  0025         UNADR = XXXXXX G    UNCOMP= XXXXXX G
UNDEF =  0080         VALTG = XXXXXX G    VALUND=  0040         WREND = XXXXXX G    WRRS  = XXXXXX G
XAXIS = XXXXXX G      XFRIT    068DR      XFRRTN   06CBR        XFRSCN   0680RG     YAXIS = XXXXXX G
ZX    = XXXXXX G
  ABS.    0000      00
          08C5      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  1380.  WORDS
.SY:MTCTL/C(DK1:SEICL),MTCTL
```

CROSS REFERENCE TABLE (CREF VM01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

```
FILEND  1-302#  7-145%  8-37%   8-66    8-87
FILINM  27-53#  27-62
FILINN  27-45#  27-142  27-145
FILLOC  1-303#  2-79    2-83%   2-104   2-149%  2-162%  8-38    8-49%   8-67%   8-89%   9-38%   12-52%
FILNAM  27-51   27-56#
FIX1    1-297#  7-61    8-81    8-97    26-15#  27-64
FIX1B   1-117#  7-62    7-64    8-99    27-65
FIX1B   1-118#  7-63    8-86
FIXNUM  1-294#  7-20    8-28    20-11
FMTVLD  1-203#
FNDOK   8-89#
FNFLG   1-27#
FORTG   1-126#
FPAITT  28-3#   28-45
FPC     1-312#  8-80    8-96    28-34   28-39%
FRET    14-3#   14-17%  14-19%  14-30
GETLAR  26-7#   27-182
GLBFLG  1-31#   27-31
GOSIG   1-125#
IDLE    27-36   27-71#
IMXTG   1-129#
INAITT  1-299#  7-106   7-113
INITMT  22-11   23-6#   23-8#   28-21
INPUTK  1-219#
INT     7-3#    7-58
IOBFR1  1-266#
IOCLNR  2-7#    2-19    26-11#  27-226  28-63
IOFLGS  1-246#  20-30%
IOFUNC  1-190#  2-16%   19-52   19-73   27-173  27-232
IOSCAN  2-5#
ITM1TG  1-136#
ITM2TG  1-137#
ITTDEV  1-260#
JMPX    1-55#
KBDEV   1-255#  27-137
KBFLAG  1-210#
KBIN    1-208#
KEYFLG  1-35#
KEYSTK  1-52#
KIFILE  3-21#
KILL    1-277#  3-12#
KILTYP  1-306#  3-9#    3-12
LBRKTG  1-139#
LCLFLG  1-24#
LDAX    1-65#
LDXX    1-60#
LENGTH  1-240#  28-15%  28-22   28-51
LISSOM  27-175  27-177#  27-249
LIST    28-3#   28-8
LISTTG  1-127#
LITCNT  27-148  27-152  27-155#
LNNOTG  1-135#
LOCTG   1-145#
LSP     1-46#
LSTABT  27-218#
LSTAGN  27-201  27-204  27-207  27-228#
LSTEXT  27-112  27-213#  27-230  27-243
```

CROSS REFERENCE TABLE (CREF V01-03)

```
LSTFIX  27-213#
LSTFMT   1-227#  27-235
LSTFNC  26-19#   27-233
LSTSND  27-234   27-236#
LSXNRM  27-190   27-200   27-203   27-211#  27-245
M.11    23-10    23-26    23-38#
M.22    23-35    23-39#
MAGEND   1-314#  24-3
MARKS    1-287#   7-146
MKFIL    7-85     7-87#    7-161
MKFILE   7-76#    7-77#
MKHDR    7-89     7-96#
MODLOP   5-16#    5-25
MODMTH   1-311#   5-8#     5-11#    6-24     6-28    18-66    18-94
MODT88   5-8#     5-9#     6-20     7-99     7-127   18-62    18-90
MRFLX    7-55     7-57#
MRKDNN   7-155    7-162#
MRKFIL   7-92     7-119#
MRKIT    7-17     7-33#
MRKITT   7-121    7-125    7-128#
MRKOK    7-158    7-161#
MT.1     8-82     8-86#
MT.256   8-63     8-72#
MT.ERF   8-83#    8-88     8-98
MTAO     8-39     8-54#
MTAOX    8-53#    8-55     8-60     8-74
MTAOXX   8-52#    8-85
MTBFIN   8-10#    8-30     8-34#    28-23
MTASCI   8-104    8-113#
MTBFR    1-267#   1-305#   7-107    7-114   14-22    14-26    14-28    28-29    28-31    28-41    28-46    28-48
MTBGFL   12-9#    12-39
MTBINT   8-130    12-47    14-2#    14-6#    19-38
MTBINY   8-106    8-111#
MTBSWP   11-15    13-46    14-4#    14-16#   19-36
MTCDON   11-8     11-22#
MTCHKS   1-288#   7-133    9-47     12-27    15-6#
MTCKIT   9-41     9-47#
MTCKSM   1-351#   7-135    9-40     12-29
MTCLOS   1-293#   7-164    8-35     11-2#    11-6#    27-224
MTDATM   18-17    18-62#
MTERR    9-8#     9-11*    9-15*
MTEXX    16-32    16-35    16-37#
MTEXXK   18-11    18-56    18-72#   18-84
MTEXXQ   18-66#   18-104
MTEXXR   18-6     18-50    18-69    18-73#   18-78
MTEXXZ   18-67#   18-100
MTFDON   8-71     8-122    8-124#
MTFFST   1-342#   1-344#   7-81     8-68     9-28     9-36     11-23    12-35    12-50    13-27
MTFIND   1-309#   6-6      8-28#
MTFLEN   1-350#   7-48     8-125
MTFLGS   12-8#    12-20*   12-40*
MTFPOS   1-325#   1-329#   7-39     18-10    18-55    18-83    25-22
MTFRD    8-50     8-57     8-61#
MTFSIZ   1-325#   1-327#   8-127
MTFSRC   8-43     8-58#
MTKEXX   6-8      6-11     6-17#
MTKHDR   6-13#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
MTKILL  1-308#  6-6#
MTKOK   6-14    6-18#
MTM1    7-24    7-27#
MTM2    7-26#   7-37    7-42
MTM3    7-21    7-22    7-25#   7-78
MTMARK  1-367#  7-16#
MTMOX   1-292#  7-139*  9-21*   12-33*  13-44   14-9*   19-61
MTMOK1  7-40    7-43#
MTNHDR  1-352#  6-10    7-88    7-124   7-142   8-62    18-68   28-13
MTNULL  1-301#  7-91    7-97    10-5#   10-6#
MTOPEN  1-326#  1-331#  8-128   16-26
MTP1    18-34   18-35   18-38#
MTPADR  22-5#   22-11#
MTPD2   1-259#  25-19   27-219
MTPDEV  1-257#
MTPIN   19-18#  19-23#  19-70   19-82
MTPINP  18-33#  22-30
MTPINR  16-22#  17-4#   18-33   18-42
MTPINW  16-21#  16-24#  17-4    18-5    18-21   18-49   18-77
MTPOLD  18-21#  22-28
MTPOT   13-16   13-25   13-29#  13-60
MTPOUT  12-1#   13-13#  13-14#
MTPPRI  18-49#  22-27
MTPRD   18-42#  22-29
MTPRG   8-117   8-119#
MTPRG.  8-108   8-115#
MTPRGT  8-110   8-112   8-114#
MTPRWD  28-58   28-59#
MTPSAV  18-77#  22-32
MTPSPC  22-16   22-27#
MTPTR   1-291#  7-130*  9-17*   9-42    11-12   12-42*  13-39   13-41*  14-7*   19-50   19-59*
MTPWRI  18-5#   22-31
MTR.RD  9-13#   9-54
MTR.RR  9-12    9-16#
MTRBFR  19-18#  19-26   19-32#
MTRCON  9-46    9-50#
MTRDER  9-6#    9-55#
MTRDON  9-45    9-49    9-51    9-58#
MTREAD  8-72    9-4#    9-9#    19-37
MTRERX  16-27   16-29   17-6    17-10   17-15   17-18   18-8    18-23   18-25   18-28   18-36#  18-44   18-53   18-81
        18-86   18-88
MTSASC  1-325#  1-333#  6-13    8-70    8-113   16-34   18-59   18-97
MTSBIN  1-325#  1-334#  6-13    8-111   18-14
MTSCRT  21-2#   21-5#
MTSECT  8-118   8-120#
MTSEX   20-26#
MTSEX1  20-23   20-25#
MTSLST  1-335#  16-28   28-54
MTSNEW  1-325#  1-332#  6-13    8-109   16-31   18-13   18-58   18-96
MTSPGM  1-325#  1-336#  8-119   18-97
MTSRD   1-342#  1-346#  13-15   13-17   19-24   19-45   19-64
MTSREG  1-349#  6-9     7-47    7-87    7-123   7-134   7-141   8-61    8-124   9-39    12-28   18-67   20-14*  20-22
        20-24   20-25*  28-12
MTSSEC  1-325#  1-338#  8-123   17-17   18-85   21-6    27-79
MTSSET  20-8#   20-11#
MTSTAT  1-324#  6-12    6-30*   7-38    8-114   8-129*  11-25*  14-24   16-25   17-16   18-9    18-12   18-15*  18-54
        18-57   18-60*  18-82   18-95   18-98*  21-5    21-7*   23-41*  25-21   25-23*  27-78   28-36   28-53
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
P1QLY   9-30   13-23   22-6#   23-15*
P1AMTA  7-33    9-14#  12-21   22-7#   23-2#   23-27   23-29*  23-31*
P1AMTB  22-7#  24-10
PLOSTG  1-123#
PNDEOF  1-45#  11-10   11-21*  13-32*  13-52   19-95*  27-107* 27-241
PNOFLG  1-44#
PNTNTG  1-132#
PNTSTG  1-130#
POINT   1-241#  7-43
PPMODE  1-224#
PRGBLN  5-30#   5-35#   6-21    6-25
PRGDAT  5-30#   5-37#  18-63
PRGPGM  5-30#   5-36#  18-91
PRIDEF  1-206#
PRTTG   1-141#
PSCTG   1-131#
PSHFPN  1-295#  7-44   14-21
PULFPN  1-296#  13-49  19-40
PUPTAP  7-165   9-24   12-46   22-9#   23-21
PUTBYT  26-13#
R0      1-38#   2-11*   2-14*   3-16*   4-14*   5-18*   5-23    7-105*  7-112*  8-76*   8-92*   27-118* 28-30*
R1      1-38#   5-12*   5-15*   5-19    5-22*   7-101*  7-108*  8-78*   8-94*   27-188* 28-32*  28-42*
R10     1-38#   7-128*  27-144* 27-149  27-177  27-179  27-194* 27-197  27-198  27-246  27-248*
R11     1-38#   27-147* 27-150* 27-154* 27-199  27-202  27-206* 27-244
R12     1-38#
R13     1-38#
R14     1-38#
R15     1-38#   27-143* 27-151
R16     1-38#
R17     1-39#
R18     1-39#
R19     1-39#
R2      1-38#   7-103*  7-110*  28-44*
R20     1-39#
R21     1-39#
R22     1-39#
R23     1-39#
R3      1-38#
R4      1-38#
R5      1-38#
R6      1-38#
R7      1-38#
R8      1-38#   7-27*   7-120   7-154   7-157*
R9      1-38#   7-72*    111    7-140   7-160*
RDBCRE  19-77   19-83#
RDBEN   19-85#  19-96
RDBEND  19-79   19-86#
RDBFRV  19-25   19-28   19-50#
RDEOF   19-33   19-56   19-89#
READRS  1-285#  9-22
REALLY  27-159
RECCNT  8-15#   8-65*   8-100*  9-34*   12-11   12-13*  12-48   13-20   13-22*  13-29   19-32   19-35*
RECLFG  1-217#
REWIND  1-284#  7-153   8-45    8-52    23-42
REWIL   1-152#
RMAT    1-151#
RPTCTL  1-212#
```

CROSS REFERENCE TABLE (CREF V01-03)

```
RSTR    1-150#
RTRNTG  1-140#
RUNFLG  1-32#
RUNN    1-218#
RWNDIT  8-41    8-44#
SAVFNC  26-20#  27-222
SBP     1-47#
SCALER  1-82#
SCRMS   19-19#  19-57
SECDEF  1-205#
SEMITG  1-144#
SERCHS  1-283#  8-58
SETARG  3-10#
SETBIT  20-20   20-24#
SNDBFR  1-313#  26-12#  27-212  28-50
SNDIT   1-228#
SPCIOB  2-6#    2-17
SPCIOF  1-278#
STGT32  1-247#  18-99   27-81
STKBLD  26-10#  27-121
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSERR  1-43#
T51DEV  1-261#
TABPTR  1-243#  8-34*   8-36
TAGPTR  1-239#
TAPE    1-276#  2-9#    3-20
TAPF2   4-6#    4-17    27-7#   27-119#
TAPFIL  3-7#    26-5#   27-24   27-116#
TCOL    1-245#
TLOON   28-14   28-28   28-56#
TLIST   28-5#   28-6#
TLISTT  28-15#  28-55
TODCON  27-82   27-85#
TODSEC  27-80   27-83#
TOLD    27-87#  27-106
TRANSL  27-10#  27-98
TRCFLG  1-34#
TSTSNF  8-17#   8-22
TYPARG  1-279#
TYPASC  5-30#   5-33#   18-61   18-89
TYPBIN  5-30#   5-32#   18-16
TYPEXT  27-125  27-128  27-131#
TYPLST  5-30#   5-34#   7-126
TYPNEW  5-30#   5-31#   6-19    7-98
TYPOLD  27-73#  27-176
TYPSEC  5-30#   5-38#   18-101
UNADR   2-18    26-9#   27-225
UNCOMP  26-6#   27-228
UNDEF   1-81#
VALTG   1-134#
VALUND  1-88#
WREND.  12-5#   12-45
WRRS    1-286#  12-43
XAXIS   1-248#  18-87   27-84*
XFRIT   25-12   25-17#
```

CROSS REFERENCE TABLE (CREF V01-03)

XFRRTN 25-10 25-24#
XFRSCN 22-17 25-4# 25-9# 25-16
YAXIS 1-343# 7-16 7-19* 20-9# 20-12 20-17* 20-26 27-6# 27-76* 27-105* 27-108
ZX 9-7# 9-33 27-205 27-214

CROSS REFERENCE TABLE (CREF V01-03)

SCN 22-22# 22-27 22-28 22-29 22-30 22-31 22-32
SEI 1-3#

```
MM      MM  TTTTTTTTTT  DDDDDDDDD   RRRRRRRRR   VV      VV                  LL          SSSSSSSS   TTTTTTTTTT
MMMM  MMMM  TTTTTTTTTT  DDDDDDDDDD  RRRRRRRRRR  VV      VV                  LL         SSSSSSSSSS  TTTTTTTTTT
MM MMMM MM      TT      DD      DD  RR      RR  VV      VV                  LL         SS      S       TT
MM  MM  MM      TT      DD      DD  RR      RR  VV      VV                  LL         SS              TT
MM  MM  MM      TT      DD      DD  RRRRRRRRRR  VV      VV                  LL         SSSSSSSSS       TT
MM      MM      TT      DD      DD  RRRRRRRRR   VV      VV                  LL          SSSSSSSSS      TT
MM      MM      TT      DD      DD  RR   RR      VV    VV                   LL                SS      TT
MM      MM      TT      DD      DD  RR    RR     VV    VV                   LL          S     SS      TT
MM      MM      TT      DDDDDDDDD   RR     RR     VVVV                .....  LLLLLLLLLL  SSSSSSSSS   TT
MM      MM      TT      DDDDDDDD    RR     RR      VV                 ....   LLLLLLLLLL   SSSSSSS    TT
```

TABLE OF CONTENTS

```
                        .TITLE  MTDRV----MAGTAPE HARDWARE DRIVER
                        .IDENT  /CRM022/

                ;READ AND WRITE AND PART OF MARK ROUTINES ARE NON-MASKABLE-
                ;INTERRUPT DRIVEN.
                ;THIS ROUTINE IS FOR MAKING AV A COMPATIBLE FORMAT AS 4923.
                ;THIS REQUIRES HARDWARE CHANGES ON CPU BOARD AND TAPE DRIVE BOARD

                ;8-MAY-75

                ;THE FOLLOWING PROGRAM CONCEPTS WERE DEVELOPED BY "H. IWATA"

                ;GHOST WRITING AND EDITING BY "R. MULLER"

                ;R128 IS NOT CONSIDERED.

                ;WRRS SUBROUTINE SHOULD USE 128 BYTES FLAG LATER

                ;TNIMS AT THE END OF READRS AND WRRS SHOULD BE REVIEWED.

                        .GLOBL  MTMAX,MTFLGS,ERRCD,EREOM,FILFND,MTXBYT,CNFRCS
                        .GLOBL  WREND.
                        .GLOBL  PUPTAP,REWIND,MTPTR,WRRS,SERCHS,EROVRD,EREOM
                        .GLOBL  BAKARS,READRS,MARKS,FILLOC,SKIPS
                        .GLOBL  MTFFST
                        .GLOBL  PIAHY,PIALY,PIAMTA,PIAMTB
                        .GLOBL  MTEOF,BOF,FFOT,NMISRV,BPIAMTB
                        .GLOBL  R10,EARLYW,MAGEND,CCODES,ERNCRT,TMP1
                        .GLOBL  SYSERR,DELY3S,DELY1S,PIALT,MTSIT2
                        .GLOBL  ERFNFD


0008            MARKIT=10       ;BITS FOR TRUE THAT IT WAS A MARK
0080            FFWRIT=200      ;FLAG FOR FIRST CYCLE OF REWRITE OPERATION
007F            NFFWRIT=177     ;COMPLEMENT OF FFWRIT
0001            MTEOF=1         ;FLAG FOR END OF FILE
00FD            NBOF=375        ;COMPLEMENT OF BOF
00F7            NMARK=367       ;REMOVES MARK PART OF BOF FLAG
0009            BOF=12          ;BEGIN FILE FLAG AND TELL IF MARK AND NOT WRITE
0004            FFOT=4          ;FLAG FOR FIRST FILE ON TAPE
                        .GLOBL  MTDRV
0000'           MTDRV   = .
```

```
                                        .SBTTL  PUPTAP---PIA INITIALIZATION
                                ;POWER UP PIA2.3 INITIALIZATION ROUTINE

0000G  86  16          PUPTAP:  LDA A   026,I          ;THIS IS THE POWER UP ROUTINE
0002G  B7  0001G                STA A   PIAMTB+1       ;TALKING TO CONTROL REGISTER/V
0005   86  29                   LDA A   51,I
0007G  B7  0000G                STA A   PIAMTB
000A   97  00G                  STA A   BPIAMTB,D
000C   86  12                   LDA A   022,I
000EG  B7  0001G                STA A   PIAMTB+1
0011G  B7  0001G                STA A   PIALY+1
0014G  B7  0001G                STA A   PIAMTA+1
0017   86  FF                   LDA A   377,I
0019G  B7  0000G                STA A   PIAMTB
001CG  B7  0000G                STA A   PIALY
001E   86  02                   LDA A   2,I
0021G  B7  0000G                STA A   PIAMTA          ;DDR CARTDRIDGE RESET
0024   86  16                   LDA A   026,I
0026G  B7  0001G                STA A   PIALY+1
0029G  07  0001G                STA A   PIAMTB+1
002CG  B7  0001G                STA A   PIAMTA+1
002FG  7F  0001G                CLR     PIAHY+1
0032G  7F  0000G                CLR     PIAHY
0035   86  2E                   LDA A   056,I
0037G  B7  0001G                STA A   PIAHY+1
003AG  7F  0000G                CLR     MTFLGS  ;CLEAR MAG TAPE FLAGS
003DG  7F  0000G                CLR     MAGEND
0040   39                       RTS
```

```
                                        .SBTTL  REWIND---REWIND DRIVER
                                ;'REWIND'  SUBROUTINE
                                ;TO REWIND TAPE
                                ;WHEN THE FIRST FILE IS TO BE WRITTEN ON THE TAPE,
                                ;SET "FFOT"=1 (FIRST FILE ON TAPE) IN "MTFLGS" (MAG TAPE FLAGS)
                                ; TO ERASE A PORTION OF THE TAPE
                                ;BETWEEN PHYSICAL "BOT" (BEGINNING OF TAPE) AND LOAD POINT
                                ; AND BEYOND THE MOTOR STOPS.

0041    BD    00A3'     EXITMT: JSR     SERRMT
0044    7E    0416'             JMP     RESMSK

0047    BD    0000'     REWIND: JSR     PUPTAP          ; THIS MAKES IT NICE AND CLEAN
004A    7F    0000G             CLR     MTSTT2          ; CLEAN UP MTCTL'S STATUS
004D    C6      27              LDA B   47,I
004F    F7    0000G             STA B   PIAMTB  ;REV,FAST,READ, START MOTOR
0052    BD    0094'             JSR     TNIMS   ;TNIM=1?
0055    F6    0000G             LDA B   PIAMTB  ;CLEAR FLAGS
0058    C6      29              LDA B   51,I    ;READ,NORM,FORWARD,STOP
005A    F7    0000G             STA B   PIAMTB  ;RESET STOP MOTOR F F
005D    C8      C8              EOR B   10,I    ;SET TO READ (GNFR=0)
                                                ;WRITE ZEROES ON MAGTAPE
005F    BD    0410'             JSR     SETMSK  ;MASK INTERUPTS
0062    F7    0000G             STA B   PIAMTB  ;START MOTOR
0065    F6    0000G     2$:     LDA B   PIAMTB  ;CLEAR UPPER AND LOWER HOLES FLAGS
0068    B6    0000G     3$:     LDA A   PIALT   ;CARTRIDGE PRESENT?
0068    85      08              BIT A   10,I
006D    26      D2              BNE     EXITMT  ;ESCAPE ROUTE
006F    B6    0001G             LDA A   PIAMTB+1
0072    85      C0              BIT A   300,I   ;HOLE DETECTION GOING ON
0074    27      F2              BEQ     3$
0076    CE    0209              LDX     1011,I  ;START OF 5 MILLISEC DELAY
0079    09              4$:     DEX
007A    26      FD              BNE     4$
007C    B6    0001G             LDA A   PIAMTB+1        ;READ PIAMTB+1
007F    28      E4              BMI     2$      ;IF "BOT", IGNORE IT.
0081    C6      29              LDA B   51,I
0083    F7    0000G             STA B   PIAMTB  ;STOP MOTOR
0086    BD    0416'             JSR     RESMSK  ;ALLOW INTERUPTS AGAIN WHILE
                                                ;WAITING FOR THE MOTOR TO STOP
0089    F6    0000G             LDA B   PIAMTB  ;CLEAR UPPER AND LOWER HOLES FLAGS
                                .IF EQ 0
008C    CE    0000              LDX     0,I     ;SET FILLOC=0
]]]]]
008F    DF      00G             STX     FILLOC,D
                                .ENDC
0091    7F    0000G             CLR     EARLYW  ;CLEAR EARLYWARNING FLAG
0094    B6    0000G     TNIMS:  LDA A   PIALT   ;CARTRIDGE TEST
0097    85      08              BIT A   10,I
0099    26      08              BNE     SERRMT
0098    B6    0000G             LDA A   PIAMTA  ;WARN 4NFR AND FILFOUND FLAG CLEARED
009E    85      08              BIT A   10,I
00A0    27      F2              BEQ     TNIMS   ;TAPE NOT IN MOTION TEST LOOP
00A2    39                      RTS
00A3    86      00G     SERRMT: LDA A   ERNCRT,I        ;ERROR NO CARTRIDGE PRESENT
00A5    97      00G             STA A   ERRCD,D
00A7    BD    0000'             JSR     PUPTAP
00AA    39                      RTS
```

```
58
59
60
61
```

```
                                                .SBTTL  FNFRS---FIND NFR'S
                                        ;'FNFRS'  SUBROUTINE
                                        ;"EREOM" IS SET AT END OF TAPE
                                        ;"FDPEOT" ROUTINE  -----DETECTING PHYSICAL END OF TAPE AND
                                        ; SET FLAG "EREOM" IN "ERRCD"

 00AB  B6  0000G    FNFRS:  LDA A   PIALY   ;CLEAR 8 NFR FLAGS
 00AE  B6  0000G            LDA A   PIAMTA  ;CLEAR 4NER FLAG
 00B1  8D  24               BSR     CNFRCS  ;CLEAR NFR COUNTER
 00B3  B6  0001G    TAGAIN: LDA A   PIAMTA+1        ;READ 4NFRS
 00B6  2B  2A               BMI     FOUND           ;GET RTS THERE
 00B8  B6  0000G            LDA A   PIALT           ;CHECK CARTRIDGE
 00BB  85  08               BIT A   10,I
 00BD  26  E4               BNE     SERRMT          ;SET ERROR MAG TAPE
 00BF  B6  0001G            LDA A   PIAMTB+1        ;READ UPPER OR LOWER HOLE FLAG
 00C2  2A  EF               BPL     TAGAIN
 00C4  B6  0001G    FDPEOT: LDA A   PIAMTB+1        ;READ UPPER OR LOWER HOLE
 00C7  85  C0               BIT A   300,I
 00C9  27  08               BEQ     1$      ;IF IT GOES TO 1$ ITS NOT THE END OF TAPE
 00CB  85  40               BIT A   100,I   ;EARLY WARNING HOLE ?
 00CD  27  03               BEQ     2$
 00CF  7C  0000G            INC     EARLYW  ;SET EARLY WARNING FLAG
 00D2  86  00G      2$:     LDA A   EREOM,I ;PHYSICAL END OF THE TAPE
 00D4  97  00G              STA A   ERRCD,D
 00D6  39           1$:     RTS
 00D7  96  00G      CNFRCS: LDA A   BPIAMTB,D       ;CLEAR NER COUNTER
 00D9  16                   TAB
 00DA  84  DF               AND A   337,I
 00DC  B7  0000G            STA A   PIAMTB
 00DF  F7  0000G            STA B   PIAMTB
 00E2  39           FOUND:  RTS
```

```
                                        .SBTTL  MARKS---MARK MAGTAPE DRIVER
                                        ;
                                        ;'MARKS'  SUBROUTINE
                                        ;
                                        ;ENTRY CONDITIONS (MTMAX-MTPTR)=LENGTH OF RECORD
                                        ;MTPTR=LOCATION OF FIRST VALID DATA BYTE
                                        ;MTMAX=LOCATION OF LAST VALID DATA BYTE
                                        ;FILFND=# OF RECORDS PER FILE TO MARK
                                        ;R10=0  IF LAST FILE THEN MAKE A 300 MILLSEC STRIP OF TAPE
                                        ;R10 NEQ 0 IF MORE FILES COMING
                                        ;
00E3  DE  00G     MARKS:  LDX     FILFND,D        ;NUMBER OF RECORDS/FILE TO MARK
00E5  09                  DEX
00E6  DF  00G             STX     FILFND,D        ;NEED THIS EXTRA DECREMENT FOR STEVE
                          .IF EQ 1
                          LDA A   MTFLGS,D        ;MAGTAPE FLAGS
                          BIT A   FEOT,I  ;BIT TEST ON FIRST FILE ON TAPE FLAG
                          BEQ     MORFIL
                          JSR     REWIND  ;ERASE PORTION ON TAPE
                          .ENDC
00E8  86  0G      MORFIL: LDA A   BOF,I   ;SET BOF AND FACT WE ARE MARKING
00EA  97  00G             STA A   MTFLGS,D
00EC  DE  00G             LDX     MTPTR,D
00EE  DF  00G             STX     TMP1,D  ;NEED START FOR MULTIPLE RECORDS
00F0  BD  01EC'   MORREC: JSR     MARKRS  ;CALL MARKRS
00F3  96  00G             LDA A   MTFLGS,D
00F5  84  FD              AND A   NBOF,I  ;CLEAR BOF
00F7  97  00G             STA A   MTFLGS,D
00F9  DE  00G             LDX     TMP1,D  ;NEW START FOR MULTIPLE RECORDS
00FB  DF  00G             STX     MTPTR,D ;DECREMENTED IN WRCONT
00FD  B6  0000G           LDA A   EARLYW  ;WAS EARLY WARN SET?
0100  26  09              BNE     2$      ;GET OUT IF BRANCH TAKEN
0102  DE  00G             LDX     FILFND,D        ;THE NUMBER OF RECORDS PER
                                                  ;FILE TO MARK
0104  09                  DEX
0105  DF  00G             STX     FILFND,D        ;NEW NUMBER ONE LESS
0107  27  02              BEQ     2$
0109  20  E5              BRA     MORREC  ;DO NEXT RECORD
0108  96  00G     2$:     LDA A   MTFLGS,D        ;SET EOF
010D  8A  01              ORA A   MTEOF,I
010F  97  00G             STA A   MTFLGS,D
0111  BD  01EC'           JSR     MARKRS  ;CALL MARKRS
0114  B6  0000G           LDA A   EARLYW  ;WAS IT ON?
]]]]]
0117  26  09              BNE     5$      ;DO 300 MILLISEC STRIP
0119  B6  0000G           LDA A   R10     ;IF 0 IMPLIES NO MORE FILES TO MARK
011C  27  04              BEQ     5$      ;DO 300 MILLSEC STRIP IF TRUE
011E  7F  0000G   7$:     CLR     MAGEND  ;ALL DONE ON THIS PASS-LETS DO IT AGAIN
0121  39                  RTS             ;RETURN OR UPDATE AGAIN OR JUST PLAIN GET OUT
0122  BD  0410'   5$:     JSR     SETMSK  ;SAVE CONDITION CODE STATUS
                                          ;AND SET INTERRUPT MASK ON
0125  BD  043B'           JSR     STAWRT
0128  86  FF              LDA A   377,I           ;300 MILLI-SEC DELAY CALL
012A  C6  1E              LDA B   36,I
012C  BD  0322'           JSR     DELY3S
012F  BD  0437'           JSR     STPWRT
0132  BD  0416'           JSR     RESMSK  ;RESTORE CONDITION CODES
0135  BD  0094'           JSR     TNIMS
```

```
                0138    BD    0463'           JSR     CREADS  ;RESTORE TO THE READ STATE
                0138    B6    0000G           LDA A   EARLYW
                013E    27      DE            BEQ     7$      ;GET OUT NO ERRORS
                0140    BD    0047'           JSR     REWIND  ;FORCE REWIND IF AT END OF TAPE
                0143    20      D9            BRA     7$      ;GET OUT CLEARING MAGEND
```

```
                                        .SBTTL  GENRM1&GENRM2--NFR GENERATION ROUTINES
                                ;"GENRM1" & "GENRM2" ROUTINE
                                ;"GENRM1" IS CALLED TO WRITE A FILE OR RECORD MARK
                                ;AT THE BEGINNING OF A RECORD. AFTER THIS ROUTINE
                                ;"WRDAS" ROUTINE IS CALLED TO WRITE DATA.
                                ;"GENRM2" IS CALLED TO WRITE A FILE OR RECORD MARK
                                ;AT THE END OF A RECORD AFTER THE "WRDAS" IS CALLED.

0145  86  20      GENRM2: LDA A   40,I    ;DISABLE WRITE ENABLE SIGNAL
0147  B7  0000G         STA A   PIAMTB
014A  96  00G           LDA A   MTFLGS,D
014C  85  01            BIT A   MTEOF,I
014E  26  04            BNE     5$
0150  C6  04            LDA B   4,I
0152  20  02            BRA     6$
0154  C6  08      5$:   LDA B   10,I
0156  86  14      6$:   LDA A   24,I    ;DISABLE "MORE DATA" INTERRUPT AND
                        ;                       CHANGE CB1 INPUT TO TRIGGER AT NEGATIVE EDGE
0158  B7  0001G         STA A   PIALY+1
015B  B6  0000G         LDA A   PIALY   ;CLEAR PIALY FLAG
015E  B6  0000G 1$:     LDA A   PIALT   ;CARTRIDGE TEST
0161  85  08            BIT A   10,I
0163  26  2D            BNE     CARGON
0165  B6  0001G         LDA A   PIALY+1 ;CHECK END OF DATA
0168  2A  F4            BPL     1$
016A  B6  0000G GENCON: LDA A   PIAHY   ;TRIGGER ONE SHOT
016D  86  30            LDA A   60,I    ;SET GNFR=1 AND WRITE MODE
016F  B7  0000G         STA A   PIAMTB
0172  5A                DEC B   ;ACC B HAS NUMBER OF NFRS
0173  2C  06            BGE     2$
0175  86  20            LDA A   40,I
0177  B7  0000G         STA A   PIAMTB  ;SET GNFR=0
017A  39                RTS             ;OUT IF NUMNFR<0
017B  B6  0000G 2$:     LDA A   PIAMTA  ;READ ONE SHOT OUTPUT
017E  85  01            BIT A   1,I
0180  26  F9            BNE     2$
0182  20  E6            BRA     GENCON  ;IF YES, CONTINUE WRITE RECORD MARK
0184  96  00G   GENRM1: LDA A   MTFLGS,D
0186  85  02            BIT A   2,I     ;CHECKS IF BEGINNING OF FILE
0188  26  04            BNE     2$
018A  C6  04            LDA B   4,I
018C  20  DC            BRA     GENCON
018E  C6  08      2$:   LDA B   10,I
0190  20  D8            BRA     GENCON
0192  7E  00A3' CARGON: JMP     SERRMT  ;GET RTS THERE

```

```
                                        .SBTTL  WRRS----WRITE A RECORD
                                ;'WRRS'  SUBROUTINE
                                ;ENTRY CONDITIONS: "MTFLGS" BOF AND EOF
                                ;BOF: BEGINNING OF FILE
                                ;MTEOF: END OF FILE
                                ;REWRITE A RECORD SUBROUTINE
                                ;ENTRY CONDITION: MTEOF,BOF FLAGS IN MTFLGS

 0195   96   00G     WRRS:   LDA A   MTFLGS,D        ;SET FFWRIT
 0197   8A   80              ORA A   FFWRIT,I
 0199   84   F7              AND A   NMARK,I ;IT ISN'T THE MARK ROUTINE THIS TIME
                                                ;SO IT WILL REQUIRE A DIFFERENT EXIT
                                                ;FROM THE WRITE ROUTINE
 019B   97   00G             STA A   MTFLGS,D
 019D   8D   38              BSR     INTWRT  ;ENABLE EOT INTERRUPT
                                                ;MASK HARDWARE INTERRUPT
 019F   8D   0448'           JSR     STMRDS  ;START MOTOR,READS,SLOWS
 01A2   8D   00D7'           JSR     CNFRCS  ;CLEAR NFR COUNTER
 01A5   96   00G             LDA A   MTFLGS,D        ;READ MTFLGS
 01A7   85   08              BIT A   BOF,I
 01A9   27   0E              BEQ     1$
 01AB   84   FD              AND A   NBOF,I  ;CLEAR BEGINNING OF FILE FLAG
 01AD   97   00G             STA A   MTFLGS,D
 01AF   B6   0000G           LDA A   PIALY   ;CLEAR 8 NFR FLAG
 01B2   86   1E              LDA A   36,I    ;ENABLE 8 NFR INTERRUPT
 01B4   B7   0001G           STA A   PIALY+1
 01B7   20   08              BRA     2$
 01B9   B6   0000G   1$:     LDA A   PIAMTA  ;CLEAR 4 NFR FLAG
 01BC   86   17              LDA A   27,I
 01BE   B7   0001G           STA A   PIAMTA+1        ;ENABLE 4 NFR INTERRUPT
 01C1   7E   0416'   2$:     JMP     RESMSK  ;RESTORE CONDITION CODES
                                                ;AND RETURN TO CALLING PROGRAM

```

```
                                .SBTTL  READRS--READ A RECORD
                        ;"READRS" SUBROUTINE
                        ;"EREOM" IS SET AT THE PHYSICAL END OF TAPE

01C4  8D  11    READRS: BSR     INIWRT  ;;INIT READ OR WRITE OPERATION
01C6  BD  044F'         JSR     STAMTS
01C9  BD  00D7'         JSR     CNFRCS          ; CLEAR NFR COUNTER
01CC  B6  0000G         LDA A   PIAHY   ;CLEAR READ DATA FLAG
01CF  86  2F            LDA A   #57,1   ;ENABLE READ DATA INTERRUPT
01D1  B7  0001G         STA A   PIAHY+1
01D4  7E  0416'         JMP     RESMSK  ;RESTORE COND CODES AND RETURN TO CALLING
```

```
                                        .SBTTL  RANDOM ROUTINES  INTWRT,MARKRS,WRDAS
                                ;"INTWRT" ROUTINE
                                ;INITIALIZE WRITE ROUTINE

01D7  BD  0410'         INTWRT: JSR     SETMSK  ;SAVE CONDITION CODES AND SET MASK ON
01DA  BD  0457'                 JSR     FORDS   ;WE WANT MOTOR TO RUN FORWARD---
01DD  7C  0000G                 INC      MAGEND ;TELLS STEVE THAT MAGTAPE IS NOT FINISHED
01E0  B6  0000G                 LDA A   PIALY   ;CLEARS WRITE DATA FLAG
01E3  B6  0000G                 LDA A   PIAHY   ; CLEAR READ DATA FLAG
01E6  B6  0000G                 LDA A   PIAMTB  ;CLEAR EOT FLAGS
01E9  DE    00G                 LDX     MTPTR,D ;DATA POINTER,FIRST BYTE
01EB  39                        RTS

                                ;"MARKRS" SUBROUTINE
                                ;"EREOM" IS SET AT THE END OF TAPE
01EC  8D    E9          MARKRS: BSR     INTWRT  ;INIT WRITE OR READ OPERATION
01EE  A6    00                  LDA A   0,X     ;FIRST DATA BYTE
01F0  B7  0000G                 STA A   PIALY           ;FIRST DATA BYTE
01F3  BD  0438'                 JSR     STAWRT  ;WRITE,MOTOR ON
01F6  BD  0425'                 JSR     TUTSS   ;TAPE UP TO SPEED ?
01F9  BD  0184'                 JSR     GENRM1  ;GENERATE RECORD MARKS (4 OR 8)
01FC  8D    16                  BSR     WRDAS   ;WRITE DATA SET UP
01FE  B6  0000G                 LDA A   ; 4HY   ;CLEAR READ DATA FLAG
0201  B6  0000G         1$:     LDA A   PIAHLT  ;CARTRIDGE TEST
0204  85    08                  BIT A   10,1
0206  26    09                  BNE     2$
0208  96    00G                 LDA A   BPIAMTB,D       ;LOOP WAIT FOR MOTOR STOP
020A  85    08                  BIT A   10,1
020C  27    F3                  BEQ     1$
020E  7E  0416'                 JMP     RESMSK  ;RESTORE CONDITION CODES
0211  7E  0041'         2$:     JMP     EXITMT  ;SET ERROR NO CART AND RESTORE INTERRUPTS

                                ;WRDAS ROUTINE

0214  B6  0000G         WRDAS:  LDA A   PIAHY   ;CLOCK THE ONE SHOT(40 USEC)
0217  B6  0000G         1$:     LDA A   PIAMTA  ;READ 0.S OUTPUT
021A  85    01                  BIT A   1,1
021C  26    F9                  BNE     1$
021E  B6  0000G                 LDA A   PIALY   ;CLEAR MORE DATA FLAG,CLEAR 8 NFR FLAG
                                        ;DISABLE 8 NFR INTERRUPT
0221  86    17                  LDA A   27,1
0223  B7  0001G                 STA A   PIALY+1 ;ENABLE "MORE DATA" INTERRUPT AND ALSO
0226  86    60                  LDA A   140,1   ;OUTPUT WRITE ENABLE,BPIAMTB HAS 4' STILL
0228  B7  0000G                 STA A   PIAMTB  ;NOW SET TO WRITE AND WRITE ENABLE IS SET TRUE
022B  39                        RTS
```

```
                                        .SBTTL  NMISRV--HERE'S WHERE IT ALL HAPPENS
                                ;HERE IS THE NMI INTERRUPT VECTOR.

022C  B6  0001G       NMISRV: LDA A   PIAHY+1
022F  F6  0000G               LDA B   PIAHY   ;GET READ DATA EVEN IF WE DON'T NEED IT
0232  4D                      TST A           ;RESTORE CONDITIONS CODE STATUS
0233  2B  14                  BMI     READDA  ;READ DATA IF BRANCH TAKEN
0235  96  0G                  LDA A   MTFLGS,D        ;READ MTFLGS AND CHECK FOR "FFWRIT" BIT
0237  2B  38                  BMI     NFRFND          ;GO TO NFRFND
0239  B6  0001G               LDA A   PIALY+1
023C  2B  58                  BMI     WRCONT  ;WRITE CONTINUE
023E  B6  0001G               LDA A   PIAMTA+1
0241  2B  1D                  BMI     FIREAD          ;WE GOT A 4NFR INTERUPT
0243  BD  0000'               JSR     PUPTAP
0246  BD  0000G               JSR     SYSERR
0249  DE  0G          READDA: LDX     MTPTR,D         ;NEXT DATA LOCATION
024B  9C  0G                  CPX     MTMAX,D
024D  27  05                  BEQ     TOMUCH  ;A PATCH FOR END OF READ
024F  E7  00                  STA B   0,X     ;STORE IT AT THE RIGHT PLACE
0251  08                      INX
0252  DF  0G                  STX     MTPTR,D ;LOCATION FOR NEXT TIME
0254  B6  0000G       TOMUCH: LDA A   PIAMTA  ;CLEAR 4NFR FLAG AND FILEFOUND FLAG
0257  86  17                  LDA A   27,I    ;ENABLE 4 NFR INTERRUPT
0259  B7  0001G               STA A   PIAMTA+1
025C  B6  0000G               LDA A   PIALY   ; CLEAR 8NFR BIT
025F  3B                      RTI
0260  B6  0000G       FIREAD: LDA A   PIAMTA  ;CLEAR 4NFR AND FILFOUND FLAGS
0263  86  16                  LDA A   26,I    ;DISABLE FURTHER 4 NFR INTERRUPT
0265  B7  0001G               STA A   PIAMTA+1
0268  86  2E                  LDA A   56,I
026A  B7  0001G               STA A   PIAHY+1 ;DISABLE READ DATA INTERRUPT
026D  BD  0453'               JSR     STPMTS  ;STOP MOTOR
0270  7F  0000G               CLR     MAGEND  ;ITS ALL OVER
0273  3B                      RTI
0274  B6  0000G       NFRFND: LDA A   PIAHY   ;FIRST ENTRY FOR REWRITE,CLOCK ONE SHOT
0277  B6  0000G       1$:     LDA A   PIAMTA  ;STATUS OF 40 USEC ONE SHOT
027A  85  01                  BIT A   1,I     ;40 USEC DELAY LOOP
027C  26  F9                  BNE     1$
027E  86  20                  LDA A   40,I    ;SET TO WRITE MODE, ITS FASTER THAN
                                              ;CALLING CWRITS
0280  B7  0000G               STA A   PIAMTB  ;THE TIME FROM THE 4 OR 8 NFR TO HERE SHOULD
                                              ;BE LESS THAN 60 US
0283  96  0G                  LDA A   MTFLGS,D        ;CLEAR FFWRIT
0285  84  7F                  AND A   NFFWRIT,I
0287  97  0G                  STA A   MTFLGS,D
0289  DE  0G                  LDX     MTPTR,D ;DIFFERENT CONDITIONS UNDER REWRITE-MUST DECREMENT IT FIRST
028B  A6  00                  LDA A   0,X     ;FIRST DATA TO BE REWRITTEN
028D  B7  0000G               STA A   PIALY
0290  BD  0214'               JSR     WRDAS
0293  86  26                  LDA A   26,I    ;DISABLE 4 NFR INTERRUPT
0295  B7  0001G               STA A   PIAMTA+1
0298  3B                      RTI
0299  DE  0G          WRCONT: LDX MTPTR,D     ;GET LOCATION OF POINTER
029B  9C  0G                  CPX     MTMAX,D ;IS IT THE LAST BYTE?
029D  27  0C                  BEQ     LASTDA
029F  08                      INX
02A0  DF  0G                  STX     MTPTR,D ;NEXT LOCATION
```

```
    58  02A2  A6  00              LDA A  0,X     ;WRITE ONE MORE BYTE AT PIALY
    59  02A4  B7  0000G           STA A  PIALY
    60  02A7  86  0000G           LDA A  PIALY   ;CLEAR MORE DATA FLAG
    61  02AA  3B                  RTI
    62  02AB  BD  0146'  LASTOA:  JSR    GENRM2  ;WRITE 4 OR 8 WERS
    63  02AE  96  00G             LDA A  MTFLGS,D
    64  02B0  85  08              BIT A  MARKIT,I        ;WAS IT THE MARK ROUTINE?
    65  02B2  26  1C              BNE    SAFETY  ;BRANCHS IF MARKING
    66  02B4  BD  0437'           JSR    STPWRT  ;TURN OFF MOTOR
    67  02B7  7F  0000G           CLR    MAGEND
    68  02BA  3B                  RTI
    69  02BB  8D  01      WREND:  BSR    WREND.
    70  02BD  3B          GOBACK: RTI
    71  02BE  BD  0094'   WREND.: JSR    TNIMS
    72  02C1  BD  00C4'           JSR    FDPEOT  ;RUN OFF THE END?
    73  02C4  86  16              LDA A  26,I
    74  02C6  B7  0001G           STA A  PIAMTB+1        ;DISABLE EOT INTERRUPTS
    75  02C9  BD  0463'           JSR    CREADS          ; TURN OFF WRITE MODE!!!!
    76  02CC  7F  0000G           CLR    MAGEND  ;STEVES HOLDING LOOP
    77  02CF  39                  RTS
    78  02D0  36          SAFETY: PSH A          ;SAVE STATUS OF MTFLAGS
    79  02D1  96  00G             LDA A  MTX8YT,D        ;PROPORTIONAL # 128 OR 257 BYTES
    80  02D3  BD  0320'           JSR    DELY1S  ;CALL DELAY ROUTINE
    81  02D6  BD  0437'           JSR    STPWRT
    82  02D9  32                  PUL A
    83  02DA  85  01      TSTEOF: BIT A  MTEOF,I ;END OF FILE?
    84  02DC  27  DD              BEQ    WREND   ;NOT END OF FILE YET
    85  02DE  C6  01              LDA B  1,I
    86  02E0  BD  0094'   2$:     JSR    TNIMS   ;TAPE NOT IN MOTION ROUTINE
    87  02E3  BD  0438'           JSR    STAWRT  ;START WITH WRITE ON
    88  02E6  BD  0425'           JSR    TUTSS   ;IS TAPE UP TO SPEED?
    89  02E9  BD  0437'           JSR    STPWRT  ;STOP WITH WRITING CURRENT ON
    90  02EC  5A                  DEC B
    91  02ED  27  F1              BEQ    2$
    92  02EF  20  CA              BRA    WREND   ;RETURN FOR MARKING ROUTINE ENDING
    93                                           ;F.. REWRITE SEQUENCE
```

```
                                        .SBTTL  BAKARS--BACKUP ONE RECORD
                                ;'BAKARS' 'SKIPS' SUBROUTINES
                                ;EREOM IS SET AT THE END OF TAPE
                                ;GO BACK ONE RECORD OR GO ONE RECORD FORWARD

02F1  BD  0458'   BAKARS: JSR     REVS    ;BACK
02F4  BD  044F'   SKIPS:  JSR     STAMTS  ;RUN
02F7  BD  0425'           JSR     TUTSS   ;IS TAPE UP TO SPEED
02FA  B6  0000G           LDA A   PIAHY   ;CLEAR READ DATA FLAG
02FD  B6  0000G   1$:     LDA A   PIALT   ;CARTRIGE TEST
0300  85  08              BIT A   10.1
0302  26  19              BNE     3$
0304  B6  0001G           LDA A   PIAHY+1 ;READ DATA = 1?
0307  2A  F4              BPL     1$
0309  BD  00A8'           JSR     FNFRS
030C  D6  00G             LDA B   BPIAMTB,D       ;REVERSE DIRECTION
030E  C5  04              BIT B   4.1
0310  27  04              BEQ     2$
0312  86  80              LDA A   200.1   ;6MS DEL FOR MOTOR ACCELERATION CHANGE IF REVERSE
0314  8D  0A              BSR     DELY1S
0316  BD  0453'   2$:     JSR     STPMTS  ;STOP
0319  BD  0094'           JSR     TNIMS   ;TNIM=1?
031C  39                  RTS
031D  7E  00A3'   3$:     JMP     SERRMT  ;SET CARTRIDGE ERROR AND RTS
```

```
                                        .SBTTL  DELY1S, DELY3S PROGRAMABLE MI4OUS DELAYS
                                ;MOUS HARDWARE DELAY
                                ;WARNING: WHEN THIS ROUTINE IS CALLED, THE MNFR AND FILE FOUND
                                ;FLAGS WILL BE CLEARED.
                                ;PASS # OF TIMES IN LOOP IN ACC A AND ACC B
                                ;OR IF YOU WANT TO CALL IT ONLY ONCE THEN
                                ;CALL DELY1S INSTEAD-YOU DON'T HAVE TO
                                ;SET UP ACC B THEN

0320    C6      01      DELY1S: LDA B   1,I     ;FOR THOSE WHO ONLY WANT
                                                ;TO DO IT ONCE
0322    37              DELY3S: PSH B
0323    36                      PSH A
0324    FE      00000G  1$:     LDA B   PIAHY   ;CLOCK ONE SHOT
0327    F6      00000G  2$:     LDA B   PIAMTA  ;READ PIAMTA-PD
032A    C5      01              BIT B   1,I
032C    26      F9              BNE     2$
032E    4A                      DEC A
032F    26      F3              BNE     1$
0331    32                      PUL A
0332    33                      PUL B
0333    5A                      DEC B
0334    26      EC              BNE     DELY3S
0336    39                      RTS
```

```
                                        .SBTTL  SERCHS--FAST SEARCH ROUTINE
                                ;'SERCHS' ROUTINE
                                ;STOP AND SETTING "EREOM" AT END OF TAPE
                                ;MEMORY USAGE:  FILLOC     ;PRESENT FILE LOCATION
                                                ;FILEND: FILE TO BE FOUND
                                                ;BPIAMTB: REV(PB2)
                                ;STEVE WILL NOT CALL ME IF WE ARE LOOKING FOR
                                ;FILE #1. WE ASSUME THAT IF FILE #1 IS DESIRED THAT THE
                                ;PERSON DOES A FIND 0 OR A REWIND AND DOESN'T CALL THIS ROUTINE


                                ;********* STANDARDS !!!!!!   ****************************************
                                ;FILEGAP---------
                                ;       3.27" < FILEGAP < 3.9"
                                ;       109MS <  0 IPS < 130MS
                                ;       36.3MS<  0 IPS < 43.3MS
                                ;
                                ;RECORD GAP -------------
                                ;       1.09" < REC. GAP < 1.3"
                                ;       36MS  <  0 IPS  < 43MS
                                ;
0337    BD      00A3'           EXCART: JSR     SERRMT
033A    33                              PUL B           ;CLEAN UP TH STACK
033B    39                              RTS

033C    86      04              SERCHS: LDA A   4,I             ; GET DELTA REC. COUNT
033E    36                              PSH A
033F    32                      AGAIN:  PUL A                   ; TEST IF DONE A DELTA SERCH
0340    4A                              DEC A
0341    27      28                      BEQ     DONEN
0343    36                              PSH A
0344    96      01G                     LDA A   FILLOC+1,D      ; IF NOT GO ANOTHER REC.
0346    26      05                      BNE     3$
0348    7C      0001G                   INC     FILLOC+1        ; MAKE IT RIGHT
034B    20      1D                      BRA     DONEN1
034D    D6      00G             3$:     LDA B   MTSTT2,D        ; CHECK IF IN GAP
034F    2B      19                      BMI     DONEN1
0351    91      01G                     CMP A   FILFND+1,D      ; DECIDE WHICH WAY TO GO
0353    24      0A                      BCC     1$              ; LOC >= FIND
0355    BD      02F1'                   JSR     BAKGRS
0358    BD      03FC'                   JSR     TST8NF
035B    20      E2                      BRA     AGAIN
035D    20      08                      BRA     DONEN1
035F    BD      02F4'           1$:     JSR     SKIPS
0362    BD      03FC'           2$:     JSR     TST8NF          ; SEE IF GOT TO A FILE GAP
0365    20      D8                      BRA     AGAIN           ; NOPE SO TRY AGAIN
]]]]]
0367    7C      0001G                   INC     FILLOC+1
036A    31                      DONEN1: INS
0368    86      80              DONEN:  LDA A   200,I   ;SET SEARCH FILE ON PIAMTB-7
036D    BD      043F'                   JSR     STORGR  ;STORE WITH GR FIRST
0370    B6      0000G                   LDA A   PIAMTB  ;CLEARS EOT FLAGS
0373    D6      01G             1$:     LDA B   FILFND+1,D      ;READ FILFND
0375    D0      01G                     SUB B   FILLOC+1,D      ;FILFND-FILLOC=B
0377    26      05                      BNE     2$
0379    96      00G                     LDA A   MTSTT2,D        ; IF IN RIGHT PLACE THEN DON'T MOVE
037B    2A      01                      BPL     2$
```

```
 58  037D  39                    RTS
 59  037E  22  08        2$:     BHI   3$
 60  0380  50                    NEG B
 61  0381  96  00G               LDA A MTSTT2,D
 62  0383  2B  01                BMI   4$
 63  0385  5C                    INC B
 64  0386  BD  0458'      4$:     JSR   REVS    ;SET BACKWARD IF <0
 65  0389  20  03                BRA   SERFA
 66  038B  BD  0457'      3$:     JSR   FORDS   ;SET FORWARD IF >0
 67  038E  37             SERFA:  PSH B         ;PUSH B
 68  038F  BD  045F'              JSR   FASTS   ;FAST
 69  0392  BD  044F'              JSR   STAMTS  ;START MOTOR
 70  0395  BD  0425'              JSR   TUTSS   ;TUTS=1
 71  0398  B6  0000G     1$:     LDA A PIAMTA  ;CLEAR FILE FOUND FLAG
 72  039B  B6  0001G     2$:     LDA A PIAMTB+1        ;READ PIAMTB+1
 73  039E  2B  7A                BMI   OUTSER  ;GO TO "OUTSER" AT END OF TAPE
 74  03A0  B6  0000G             LDA A PIALT           ;STATUS OF CARTRIDGE TEST
 75  03A3  85  08                BIT A 10,I            ;CARTRIDGE BIT
 76  03A5  26  90                BNE   EXCART
 77  03A7  B6  0001G             LDA A PIAMTA+1        ;READ PIAMTA+1
 78  03AA  85  40                BIT A 100,I   ;FILE FND=1?
 79  03AC  27  ED                BEQ   2$
 80  03AE  33                    PUL B                 ;PUL B
 81  03AF  5A                    DEC B                 ;B-1=B
 82  03B0  37                    PSH B                 ;PUSH B
 83  03B1  C6  05                LDA B 5,I     ; A 50MS DELAY FOR END OF TAPE TEST
 84  03B3  37             7$:     PSH B
 85  03B4  86  FA                LDA A 250.,I  ; 10MS AT A TIME
 86  03B6  BD  0320'              JSR   DELY1S
 87  03B9  33                    PUL B
 88  03BA  B6  0000G             LDA A PIAMTA          ; LOOK FOR DATA EVERY 10MS
 89  03BD  85  40                BIT A 100,I
 90  03BF  27  05                BEQ   6$
 91  03C1  5A                    DEC B
 92  03C2  27  2F                BEQ   NOFIL           ; HAD NO FILE GAP
 93  03C4  20  ED                BRA   7$              ; GO FOR ANOTHER 10MS
 94  03C6  33             6$:     PUL B
 95  03C7  5D                    TST B
 96  03C8  37                    PSH B
 97  03C9  26  CD                BNE   1$      ;MORE FILES TO GO IF B>0
 98  03CB  BD  0453              JSR   STPMTS  ;STOP MOTOR
 99  03CE  96  01G               LDA A FILFND+1,D
100  03D0  97  01G               STA A FILLOC+1,D      ;FILFND=FILLOC
101  03D2  BD  0094'              JSR   INIMS   ;READ INIM
102  03D5  96  00G               LDA A BPIAMTB,D       ;READ BPIAMTB
103  03D7  88  04                EOR A 4,I     ;CHANGING DIRECTION
104  03D9  84  FD                AND A -1-2,I          ; SLOW
105  03DB  BD  0441'              JSR   STORIT
106  03DE  96  00G       3$:     LDA A BPIAMTB,D
107  03E0  85  04                BIT A 4,I             ; FOR/REV?
108  03E2  26  05                BNE   4$
109  03E4  BD  02F4'              JSR   SKIPS
110  03E7  20  03                BRA   5$
111  03E9  BD  02F1'      4$:     JSR   BAKARS
112  03EC  BD  03FC'      5$:     JSR   TST8NF
113  03EF  20  ED                BRA   3$
114  03F1  33             OUT:    PUL B
```

```
 115  03F2  39                    RTS
 116  03F3  86  000G      NOFIL:  LDA A   ERFNED,I      ;ERROR FILE NOT FOUND
 117  03F5  97  000G              STA A   ERRCD,D
 118  03F7  BD  0047'             JSR     REWIND
 119  03FA  20  E5                BRA     OUT         ;PULL B AND RTS
 120
```

```
                        .SBTTL  TST8NF--TEST FOR 8NER BIT AND SET STATUS IF IN FILE GAP
                        ;THIS ROUTINE LOOKS FOR 8NER BIT
                        ;RETURNS+2 IF NOT FOUND
                        ;RETURNS+4 IF FOUND ALSO SET FILE GAP BIT
                        ;
                        .GLOBL  TST8NF
03FC    30      TST8NF: TSX                     ; ESTABLISH NORMAL RETURN ADDRESS
03FD    EE  00          LDX     0,X
03FF    31              INS
0400    31              INS
0401    B6  0001G       LDA A   PIRLY+1         ; LOOK AT BIT
0404    85  40          BIT A   100,I
0406    27  06          BEQ     MORE
0408    B6  00G         LDA A   MTEFST,I                ; SET FILE GAP BIT
040A    97  00G         STA A   MTSTT2,D
040C    08              INX
040D    08              INX
040E    6E  00  MORE:   JMP     0,X
```

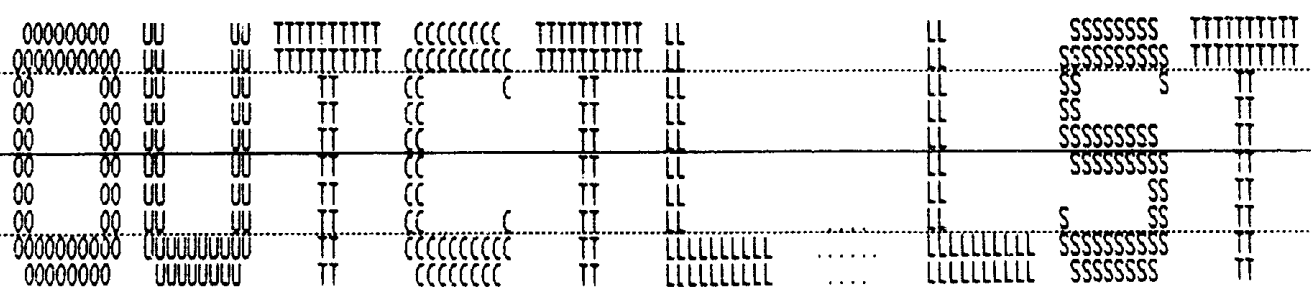
```
                                        .SBTTL  ROUTINES TO MODIFY MOTOR STATUS
                                ;SAVING CONDITION CODES AND SET INTERUPT MASK ON ROUTINE

0410    07              SETMSK: TPA     ;COND. CODES TO ACC.
0411    97      00G             STA A   CCODES,D        ;SAVE COND CODE STATUS ON ENTRY
0413                            SEI             ;SET MASK ON-DON'T ALLOW INTERUPTS
0413    01      0F      .BYTE   01,17
0415    39              RESMSK: RTS

                                ;THIS SUBROUTINE RESTORES STATUS OF THE CONDITION CODES

0416    96      00G             LDA A   CCODES,D        ;COND CODES THAT WERE SAVED
0418    06                      TAP             ;UPDATE COND CODES
0419    39                      RTS

041A    BD      0453'   OUTSER: JSR     STPMTS  ;STOP MOTOR
041D    BD      00C4'           JSR     EOPEOT  ;SET.."EREOM"=1
0420    BD      0094'           JSR     TNIMS   ;WAIT UNTIL MOTOR STOPS
0423    33                      PUL B
0424    39                      RTS

                                ;'TUTSS' ROUTINE
0425    B6      0000G   TUTSS:  LDA A   PIALT   ;CARTRIDGE TEST
0428    85      08              BIT A   10,I
042A    26      08              BNE     1$
042C    B6      0000G           LDA A   PIAMTA  ;WORN WHER,FILFOUND FLAGS ARE CLEARED
042F    85      04              BIT A   4,I     ;TAPE UP TO SPEED ?
0431    27      F2              BEQ     TUTSS
0433    39                      RTS
0434    7E      00A3'   1$:     JMP     SERRMT  ;SET ERROR NO CARTRIDGE

0437    86      28      STPWRT: LDA A   50,I    ;STOP WITH WRITING CURRENT ON
0439    20      06              BRA     STORIT
043B    86      20      STAWRT: LDA A   40,I    ;START WRITING
043D    20      02              BRA     STORIT
043F    9A      00E     STORDR: ORA A   BPIAMTB,D
0441    97      00G     STORIT: STA A   BPIAMTB,D
0443    B7      0000G           STA A   PIAMTB
0446    39                      RTS
0447    94      00G     STOAND: AND A   BPIAMTB,D
0449    20      F6              BRA     STORIT
044B    86      21      STMRDS: LDA A   41,I    ;SLOW,FORWARD,READ,START MOTOR
044D    20      F2              BRA     STORIT
044F    86      F7      STAMTS: LDA A   367,I
0451    20      F4              BRA     STOAND
0453    86      08      STPMTS: LDA A   10,I
0455    20      E8              BRA     STORDR
0457    86      FB      FORDS:  LDA A   373,I
0459    20      EC              BRA     STOAND
045B    86      04      REVS:   LDA A   4,I
]]]]]
045D    20      E0              BRA     STORDR
045F    86      02      FASTS:  LDA A   2,I
0461    20      DC              BRA     STORDR
0463    86      01      CREADS: LDA A   1,I
0465    20      D8              BRA     STORDR
                0001'           .END
```

```
MTDRV----MAGTAPE HARDWARE DRIVE       RT-11 MMAC  VM02-10   14-OCT-76 01:37:08 PAGE 15+
SYMBOL TABLE

AGAIN    033FR       BAKARS   02F1RG      BOF   =   000A G      BPIAMT= XXXXXX G      CARGON   0192R
CCODES= XXXXXX G     CNERCS   000ZRG      CREADS   0463R       DELY1S   0320RG       DELY3S   0322RG
DONEN    036BR       DONEN1   036AR       EARLYW= XXXXXX G     EREOM = XXXXXX G      ERFNFD= XXXXXX G
ERNCRT= XXXXXX G     EROVRD= XXXXXX G     ERRCO = XXXXXX G     EXCART   0337R       EXITMT   0041R
FASTS    045ER       FDPEOT   00C4R       FEOT  =   0004 G     FFWRIT=  0080        FILEND= XXXXXX G
FILLOC= XXXXXX G     FIREAD   0260R       FNFRS    00ABR       FORDS    0457R       FOUND    00E2R
GENCON   016AR       GENRM1   0184R       GENRM2   0145R       GOBACK   028DR       INTWRT   01D7R
LASTDA   029BR       MAGEND= XXXXXX G     MARKIT=  0008        MARKRS   01ECR       MARKS    00E3RG
MORE     040ER       MORFIL   00E8R       MORREC   00F0R       MTDRV =  0000RG      MTEOF =  0001 G
MTFFST= XXXXXX G     MTFLGS= XXXXXX G     MTMAX = XXXXXX G     MTPTR = XXXXXX G      MTSTT2= XXXXXX G
MTXBYT= XXXXXX G     NBOF  =  00ED        NEFWRI=  007F        NERFND   0274R       NMARK =  00F7
NMISRV   022CRG      NOFIL    03F3R       OUT      03F1R       OUTSER   041AR       PIAMY = XXXXXX G
PIALT = XXXXXX G     PIALY = XXXXXX G     PIAMTA= XXXXXX G     PIAMTB= XXXXXX G      PUPTAP   0000RG
READDA   0249R       READRS   01C4RG      RESMSK   0416R       REVS     0458R       REWIND   0047RG
R10   = XXXXXX G     SAFETY   02D0R       SERCHS   033CRG      SERFA    038ER       SERRMT   0093R
SETMSK   0410R       SKIPS    02F4RG      STAMTS   044FR       STAURT   043BR       STMRDS   044BR
STORND   0447R       STORIT   0441R       STORQR   043FR       STPMTS   0453R       STPWRT   0437R
SYSERR= XXXXXX G     TAGAIN   00B3R       TMP1  = XXXXXX G     TNIMS    0094R       TOMUCH   0254R
TSTEOF   02DAR       TST8NF   03FCRG      TUTSS    0425R       WRCONT   0299R       WRDAS    0214R
WREND    028BR       WREND    028ERG      WRRS     0195RG

. ABS.    0000     00
          0467     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2928. WORDS
.SY:MTDRV/C(DK1:SEICLI,MTDRV
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-3# 15-6



14-OCT-76

TABLE OF CONTENTS

```
                                .TITLE  OUTCTL OUTPUT CONTROL ROUTINES FOR I/O PROSESSOR
0000'                           .CSECT  OUTCTL          ; FOR SYSTEM INTERGRATION
                                .IDENT  /SBH035/
                                .GLOBL  XFRCTL          ; FIGURE OUT DEVICE DRIVER
                                .GLOBL  OUTBFR
                                .GLOBL  PRISTG          ; PRINT A STRING
                                .GLOBL  CRLF            ; PRINT A CRLF (CR)
                                .GLOBL  CRLFLF          ; PRINT A CR,LF,CR,LF (CR,CR)
                                .GLOBL  PRIVAL          ; PRINT A VALUE
                                .GLOBL  PUTBYT          ; ENTER A BUT INTO OUTPUT BUFFER
                                .GLOBL  DEFPNT          ; PRINT ROUTINE PROCESSOR
                                .GLOBL  PSHFPN          ; PUSH FPN
                                .GLOBL  A8X,A6X         ; TO ADD TO INDEX REG.
                                .GLOBL  SETBNK
                                .GLOBL  YAXIS
                                .GLOBL  FMTPNT          ; FORMATED PRINTING
                                .GLOBL  ISP,ISL         ; IMAGE POINTERS
                                .GLOBL  STAT37          ;
```

```
                                        .SBTTL  PRISTG  PRINT A STRING (USING DEFAULT ONLY FOR NOW)
                                ;THIS ROUTINE SETS UP THE NECESSARY POINTERS FOR THE PRINT FORMAT
                                ;       PROCESSOR AND THEN PROCEEDS TO CALL IT.
0000  86  000G        PRISTG: LDA A   RTRNTG,I        ; TAG IT FOR PAGE FULL
0002  36                      PSH A
0003  DE  000G                LDX     POINT,D         ; GET POINTER TO STRING
0005  DF  000G                STX     DP,D            ; FOR PRINT
0007  DE  000G                LDX     LENGTH,D        ; GET LENGTH OF STRING
0009  DF  000G                STX     DL,D            ; FOR PRINT
000B  86  02                  LDA A   2,I             ; MARK IT A STRING FOR PRINT
000D  97  000G                STA A   DT,D
000F  7E  0069'               JMP     PRISPC          ; SPECIAL PRINT TESTS
```

```
                                        .SBTTL  SOME RANDOM ROUTINES
                                ;CRLFLF OUTPUT TWO "CR" TO THE I/O DEVICE
                                        .GLOBL  IECRLF
                                ;
0012    8D      00              CRLFLF: BSR     CRLF
0014    86      0D              CRLF:   LDA A   CR,I            ; SEND "CR"
0016    BD      0009'                   JSR     PUTBYT
0019    39                      1$:     RTS
```

```
                                        .SBTTL  PRIVAL PRINT A VALUE (USING ONLY DEFAULT FOR NOW)
                                ;THIS ROUTINE PUSHES THE VALUE ON THE STACK AND CALLS FPNOSC (FPN->ASCII)
                                ;       AND CHECKS IF THE NUMBER IS VALID.  IF NUMBER IS NOT VALID IT
                                ;       SETS AN ERROR (THIS NOT IMPLIMENTED YET!)<<<<xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                ;       IT WILL CALL PRINT PROCESSOR IF ALL IS OK.  THE
                                ;       POINTER TO THE VALUE IS MOVED TO THE NEXT VALUE SO AS TO BE
                                ;       READY FOR A MATRIX
                                ;
                                ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                ;       UPDATE THE POINTER
                                        .GLOBL  FPNOSC
001A  86  00G     PRIVAL: LDA A   RTRNTG,I        ; TAG IT
001C  36                  PSH A
001D  86  01              LDA A   1,I             ; SET UP TO DO A NUMBER
001F  97  00G             STA A   DT,D
0021  7E  0069'           JMP     PRISPC          ; CHECK FOR SPECIAL PRINTS FIRST
```

```
                                        .SBTTL  PRIDFT--DEFAULT PRINT ";" ETC.
                                ; THE DEFAULT PRINT ALGORITHM
                                ; IF NO SEMICOLON DO A TAB TO THE NEXT COLUMN.
                                ; IF A SEMICOLON;
                                ;   A) IF A LITERAL WAS PRECEEDING GO DIRECT TO OUTPUT.
                                ;   B) IF NO LITERAL WAS PRECEEDING OUTPUT A SPACE
                                ;      AND THEN DO OUTPUT.
                                ;
                                        .GLOBL  PRIDFT          ; DEFAULT PRINT GOOODIY
                                        .GLOBL  EOLCHR
                    0020                SPACE   = H20
                    0000                CR      = H00
                    0020                LASTYP  = H20           ; SET IF LAST TYPE STRING
                    0002                SEMIC   = H02
                                ;
0024    86      00G     PRIDFT: LDA A   RTPNTG,I        ; TAG IT
0026    36                      PSH A
0027    96      00G             LDA A   IOFLGS,D        ; GET FLAGS
0029    85      02              BIT A   SEMIC,I         ; TEST FOR SEMICOLON
002B    27      06              BEQ     TAB             ; IF NO SEMICOON THEN TAB
002D    85      20              BIT A   LASTYP,I        ; ELSE WAS VALUE PRECEEDING?
002F    27      2C              BEQ     SP.OUT          ; YES SO SPACE FIRST.
0031    32                      PUL A
0032    39                      RTS                     ; IF STRING THE NO SPACE
                                ; THE TAB ALGORITHM
0033    DE      00G     TAB:    LDX     A.END,D         ; IF "CR" IN BUFFER THEN
0035    E6      00              LDA B   0,X             ;
0037    D1      00G             CMP B   EOLCHR,D
0039    26      05              BNE     1$
003B    BD      0110'           JSR     OUTBFR          ; OUTPUT THE BUFFER
003E    20      1D              BRA     SP.OUT
0040    B6      0000G   1$:     LDA A   TABCNT          ;GET NUMBER OF CHARS
0043    4A                      DEC A
0044    27      17              BEQ     SP.OUT          ; GO OUTPUT SPACE AND ITEM.
0046    81      37              CMP A   55.,I           ; MORE THAN 55 CHAR. THIS LINE?
0048    24      10              BCC     OUTLIN          ; IF SO OUTPUT THIS LINE AND CONTINUE
004A    80      12      TABLOP: SUB A   18.,I           ; SEARCH FOR NEAREST TAB
004C    22      FC              BHI     TABLOP
004E    27      00      SPALOP: BEQ     SP.OUT          ; GO OUTPUT SPACE AND ITEM.
0050    36                      PSH A                   ; SAVE COUNT
0051    86      20              LDA A   SPACE,I         ; OUTPUT A SPACE
JJJJJ
0053    BD      0009'           JSR     PUTBYT          ; I DON'T THINK I CAN GET PAGE FULL???
0056    32                      PUL A                   ; RETRIEVE COUNT
0057    4C                      INC A                   ; INCREMENT IT
0058    20      F4              BRA     SPALOP          ; JUMP BACK TO TEST
005A    BD      0014'   OUTLIN: JSR     CRLF
005D    96      00G     SP.OUT: LDA A   DT,D            ; IF STRING THEN NO SPACE
005F    4A                      DEC A
0060    26      05              BNE     TB.OUT
0062    86      20              LDA A   SPACE,I
0064    BD      0009'           JSR     PUTBYT
0067    32              TB.OUT: PUL A
0068    39                      RTS
```

```
                                        .SBTTL  PRISPC -- SPECIAL PRINT CASES
                                ; THIS ROUTINE CHECKS FOR SPECIAL
                                ; PRINTS TO INTERNAL DEVICED
                                ; DISPLAY SPECIAL FUNCTIONS:
                                ;       DRAW 20.
                                ;       MOVE 21.
                                ;  "ALPHAFONT" 18.
                                ; MAGTAPE SPECIAL FUNCTIONS:
                                ;       MARK 7.
                                ;       KILL 28.
                                ;       FIND 22.
                                ; WHEN CALLED THE DATA TYPE
                                ; WILL BE SET UP IN DT.
                                ;       DT=1 IF VALUE
                                ;       DT=2 IF STRING
                    0025                CPUDEV  = 37.
0069    96  000G    PRISPC: LDA A   A.PRIM,D        ; IS PRIMARY A SPECIAL DEVICE
0068    D6  000G            LDA B   A.SEC,D         ; GET SECONDARY ALSO
006D    81  20              CMP A   DSPDEV,I        ; IS IT DISPLAY
006F    27  25              BEQ     DSPSPC
0071    81  21              CMP A   MTPDEV,I        ; IS IT MAGTAPE
0073    27  26              BEQ     MTPSPC
0075    81  25              CMP A   CPUDEV,I        ; TEK51?
0077    27  18              BEQ     CPUSPC
                        ;ALL IS OK.
0079    96  000G    SPCNOT: LDA A   DT,D            ; IF VALUE THEN STILL NEED TO FLOAT IT
0078    4A                  DEC A
007C    26  08              BNE     1$
007E    DE  000G            LDX     POINT,D         ; GET NUMBER ON TO STACK
0080    8D  0000G           JSR     PSHFPN
0083    8D  0000G           JSR     FPNASC          ; AND CONVERT IT
0086    96  000G    1$:     LDA A   A.STAT,D        ; IS AN IMAGE DEFINED?
0088    85  08              BIT A   FMTVLD,I
008A    27  24              BEQ     SPCDFT
008C    8D  0000G           JSR     FMTPNT          ; THEN GO USE FORMATED PRINT
008F    31                  INS
0090    39                  RTS
0091    CE  0002'   CPUSPC: LDX     CPUSTB,I        ; CPU TABLE
0094    20  08              BRA     SPCSCN
0096    CE  0088'   DSPSPC: LDX     DSPSTB,I        ; GET DISPLAY TABLE
0099    20  03              BRA     SPCSCN          ; GO SCAN FOR SPECIAL FUNCTION
0098    CE  00C5'   MTPSPC: LDX     MTPSTB,I        ; GET MAGTAPE TABLE
009E    A6  00      SPCSCN: LDA A   0,X
00A0    2B  D7              BMI     SPCNOT          ; NOT SPECIAL FUNCTION
00A2    11                  CBA                     ; SPECIAL FUNCTION REQUIRED
00A3    27  05              BEQ     SPCFND          ; GO DO SPECIAL FUNCTION
00A5    08                  INX                     ; MOVE TO NEXT TABLE ENTRY
00A6    08                  INX
00A7    08                  INX
00A8    20  F4              BRA     SPCSCN          ; TRY THIS ONE
00AA    EE  01      SPCFND: LDX     1,X             ; GET ADDRESS
00AC    AD  00              JSR     0,X             ; GO DO SPECIAL FUNCTION
00AE    31                  INS                     ; RETURN TO CALL-1 RETURN
00AF    39                  RTS
00B0    BD  0024'   SPCDFT: JSR     PRIDFT          ; DO DEFAULT PRINT
00B3    BD  0000G           JSR     DEFPNT
00B6    31                  INS
```

```
    58        0087    39                    RTS
```

```
                                .SBTTL  SPECIAL PRINT'S FUNCTION TABLE
                                ;
                                ; THE SPECIAL FUNCTION TABLE
                                ;  1 BYTE - SECONDARY ADDRESS REQUIRING FUNCTION
                                ;  1 WORD - ADDRESS OF FUNCTION SERVICE ROUTINE
                                ; LAST BYTE = 255
                                        .MACRO  SPC     SA,FUNC
                                        .GLOBL  FUNC
                                        .BYTE   SA
                                        .WORD   FUNC
                                        .ENDM
                                ;
0088                            DSPSTB: SPC     20.,DSDRAW      ; DRAW
0088                                    SPC     21.,DSMOVE      ; MOVE
008E                                    SPC     18.,DSFONT      ; ALPHAFONT
00C1                                    SPC     2b.,DSFULL      ; PAGE FULL FUNCTION
00C4    FF                              .BYTE   255.
00C5                            MTPSTB: SPC     7.,MTKILL       ; KILL
00C8                                    SPC     28.,MTMARK      ; MARK
00CB                                    SPC     27.,MTFIND      ; FIND
00CE                                    SPC     0,MTSSET        ; MAGTAPE STATUS DRIVERS
0001    FF                              .BYTE   255.
00D2                            CPUSTB: SPC     0.(PSSET        ; SET  MODE CPU DELIMITERS
00D5                                    SPC     26.,CPSIEC      ; CR OR CR/LF FORMAT SET.
00D8    FF                              .BYTE   255.
```

```
                                        .SBTTL  PUTBYT PUT A BYTE INTO THE OUTPUT BUFFER
                                ;THIS ROUTINE ENTERS A CHARACTER IN TO THE OUTPUT BUFFER
                                ;       IF THE BUFFER WAS FULL THE BUFFER IS TRANSMITTED FIRST
                                ;       IF "CR" WAS PREVIOUS CHARACTER THEN THE BUFFER IS OUTPUT
                                ;       BEFORE THE NEW CHARACTER IS ENTERED.  THIS WAS ADDED TO MAKE IT
                                ;       POSSIBLE TO EOI (IEC) THE RIGHT WAY.
                                ;
                                        .GLOBL  ERCTRA,ERCTRB
                                        .GLOBL  SCRTCH
                                        .GLOBL  TABCNT
                                ;
0009                            PUTBYT:
0009  F6  0000G                 PUTBT.: LDA B   ERCTRA          ; FOR UNCOMP.
000C  27  03                            BEQ     1$
000E  7C  0000G                         INC     ERCTRB
00E1  DE  0CG                   1$:     LDX     A.END,D         ; TRYING TO OVER-RUN?
00E3  9C  00G                           CPX     A.MAX,D
00E5  27  06                            BEQ     2$              ; GO OUTPUT THE BUFFER
00E7  E6  00                            LDA B   0,X             ; LOOK AT LAST CHARACTER
00E9  D1  00G                           CMP B   EOLCHR,D        ; WAS IT "CR"
00E8  26  04                            BNE     EXQ             ; IF SO OUTPUT BUFFER
00ED  8D  21                    2$:     BSR     OUTBFR
00EF  DE  00G                           LDX     A.END,D
00F1  D6  00G                   EXQ:    LDA B   ERRCD,D         ; TEST FOR ERRORS
00F3  26  1A                            BNE     EXIT
00F5  08                                INX                     ; MOVE POINTER TO NEXT POSITION
00F6  DF  00G                           STX     A.END,D         ; SAVE IT
00F8  A7  00                            STA A   0,X             ; ENTER IT INTO THE BUFFER
00FA  81  00                            CMP A   15,I            ; IF "CR" RESET TABCNT
00FC  26  05                            BNE     2$
00FE  C6  01                            LDA B   1,I             ; RESET TAB POSITION FOR J.K.
0100  F7  0000G                         STA B   TABCNT
0103  81  1F                    2$:     CMP A   37,I            ; IF CONTROL CHAR. IT DOESN'T COUNT
0105  23  08                            BLS     EXIT
0107  7C  0000G                         INC     TABCNT          ; INC. TAB POSITION FOR J.K.
010A  26  02                            BNE     EXIT
010C  7C  0000G                         INC     TABCNT          ; MUST GO FROM 255-->1
010F  39                        EXIT:   RTS                     ; RETURN
                                ;*******************************************************************
                                ;       OUTPUT THE BUFFER
0110  36                        OUTBFR: PSH A                   ; SAVE PENDING CHARACTER
0111  86  00G                           LDA A   BAKSTG,I        ; A GOOD 4 BYTE TAG
0113  36                                PSH A
0114  96  00G                           LDA A   NOOUT,D         ; TEST IF OK TO OUTPUT BUFFER
0116  85  01                            BIT B   NOWRIT,I
0118  27  07                            BEQ     OKSND
011A  C6  00G                           LDA B   ERFBER,I        ; IF NOT SET ERROR-OVER-RUN
011C  D7  00G                           STA B   ERRCD,D
011E  31                        EXQ.2:  INS                     ; RETURN
011F  32                                PUL A                   ; GET CHARACTER BACK
0120  39                                RTS
0121  DE  00G                   OKSND:  LDX     A.PTR,D         ; TEST FOR EMPTY BUFFER
0123  09                                DEX
0124  9C  00G                           CPX     A.END,D
0126  27  F6                            BEQ     EXQ.2           ; THEN DONE
0128  85  02                            BIT A   LSTFMT,I        ; IF LIST FORMAT THEN EXPAND IT
012A  27  30                            BEQ     OKSNDT
```

PUTBYT PUT A BYTE INTO THE OUTPUT BUFFER

```
 58  012C CE 0002G          LDX   SCRTCH+2,I    ; MOVE LINE TO SCRATCH AREA
 59  012F FF 0000G          STX   SCRTCH
 60  0132 DE 00G            LDX   A.PTR,D       ; CONVERT ALL CONTROL CHAR. TO BS
 61  0134 A6 00     OKLOOP: LDA A 0,X           ; GET CHAR.
 62  0136 FE 0000G          LDX   SCRTCH        ; RETRIEVE RESULT POINTER
 63  0139 85 E0             BIT A 224.,I        ; TEST FOR CONTROL CHAR.
 64  013B 26 10             BNE   OKSVIT
 65  013D 81 00             CMP A 13.,I         ; TEST FOR CR
 66  013F 27 0C             BEQ   OKSVIT
 67  0141 8A 40             ORA A 64.,I         ; MAKE IT ALPHA
 68  0143 A7 00             STA A 0,X           ; SAVE IT
 69  0145 08                INX
 70  0146 86 08             LDA A 8.,I          ; PRINT BACK SPACE
 71  0148 A7 00             STA A 0,X
 72  014A 08                INX
 73  014B 86 5F             LDA A 95.,I         ; PRINT UNDERLINE
 74  014D A7 00     OKSVIT: STA A 0,X           ; SAVE CHAR.
 75  014F 08                INX                 ; MOVE POINTER
 76  0150 FF 0000G          STX   SCRTCH        ; SAVE POINTER
 77  0153 DE 00G            LDX   A.PTR,D       ; GET REAL BUFFER POINTER
 78  0155 9C 00G            CPX   A.END,D       ; DONE?
 79  0157 27 05             BEQ   OKDONE
 80  0159 08                INX
 81  015A DF 00G            STX   A.PTR,D       ; UPDATE THE POINTER
 82  015C 20 D6             BRA   OKLOOP        ; AND LOOP TIL DONE
 83  015E CE 0002G  OKDONE: LDX   SCRTCH+2,I    ; SET UP EXPANDED BUFFER POINTERS
 84  0161 DF 00G            STX   A.PTR,D
 85  0163 FE 0000G          LDX   SCRTCH
 86  0166 09                DEX
 87  0167 DF 00G            STX   A.END,D
 88  0169 96 00G    OKSNOT: LDA A IOFUNC,D      ; IS IT SAVE?
 89  016B 81 01             CMP A 1,I
 90  016D 26 07             BNE   OKK
 91  016F 96 00G            LDA A STAT37,D      ; ?
 92  0171 27 03             BEQ   OKK
 93  0173 BD 0269'          JSR   SCRMO         ;
 94  0176 96 00G    OKK:    LDA A IOFUNC,D      ; WRITE?
 95  0178 81 0F             CMP A 15.,I         ;
 96  017A 27 14             BEQ   OKKK
 97  017C DE 00G            LDX   A.END,D
 98  017E A6 00             LDA A 0,X
 99  0180 81 0D             CMP A 15,I
100  0182 26 0C             BNE   OKKK
101  0184 B6 0000G          LDA A IECRLF
102  0187 27 07             BEQ   OKKK
103  0189 86 0A             LDA A 12,I
104  018B 08                INX
105  018C A7 00             STA A 0,X
106  018E DF 00G            STX   A.END,D
107  0190 CE 018A'  OKKK:   LDX   OUTTBL,I      ; GET DISPATCH TABLE ADDRESS
108  0193 BD 01C0'          JSR   XFRCTL        ; GO PREFORM I/O FUNCTION
109  0196 DE 00G            LDX   A.STRT,D      ; RESET SOME POINTERS
110  0198 DF 00G            STX   A.PTR,D       ; SET UP POINTER FOR OUTPUT!!!!!
111                                             ; THIS NEEDS TO BE HERE!!!!!!!!!
112  019A 09                DEX
113  019B DF 00G            STX   A.END,D
114  019D 7E 011E'          JMP   EXQ.2
```

```
                                        .SBTTL  SNDBFR--SEND BUFFER WITH EOI UPON THE END
                                ;THIS ROUTINE OUTPUTS THE BUFFER IN MEMORY AND RESETS THE I/O POINTERS
                                ;TO THE BEGINNING TO ALLOW FOR MORE OUTPUT.  ALSO EOI WILL BE SENT WITH
                                ;THE LAST CHARACTER IN THE BUFFER
                                ;
                                        .GLOBL  SNDBFR
                                ;
01A0    06      000     SNDBFR: LDA B   A.SEC.O         ;
01A2    C1      10              CMP B   29.,I
01A4    26      08              BNE     2$
01A6    96      000             LDA A   A.PRIM.O
01A8    81      25              CMP A   37.,I           ;
01AA    26      05              BNE     1$
01AC    86      FF              LDA A   255.,I
01AE    97      000             STA A   STAT37.O        ;
01B0    39                      RTS
01B1    96      000     1$:2$:  LDA A   NOOUT.O         ; SET EOI SEND BIT
01B3    8A      80              ORA A   SNDIT.I
01B5    97      000             STA A   NOOUT.O
01B7    7E      0110'           JMP     OUTBFR
```

```
                                        .SBTTL  OUTPUT DISPATCH TABLE
                                        ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                        ;       DISPATCH TABLE
                                        .GLOBL  FILEOT,IECOUT,DSPOUT,MTPOUT,NIODEV
01BA    G000G                   OUTTBL: .WORD   FILEOT          ; FILE OUTPUT
01BC    0000G                           .WORD   IECOUT          ; IEC PORT OUTPUT
01BE    01C8'                           .WORD   NIODEV          ; KEYBOARD OUTPUT ?
01C0    0000G                           .WORD   DSPOUT          ; DISPLAY OUTPUT
01C2    01CC'                           .WORD   NOACT           ; MAGTAPE OUTPUT xxxxxx PATCH TO FIX PRI  3,27:5 xxxxxx

01C4    01C8'                           .WORD   NIODEV          ; OUTPUT TO DATA FIELD ?
01C6    0000G                           .WORD   MTPOUT          ; MAGTAPE OUTPUT DEVICE 2


                                        ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
01C8    86      000             NIODEV: LDA A   ERN,0D,I      ; ILLEGAL I/O OPERATION
01CA    97      000                     STA A   ERRCD,D       ; ON THIS DEVICE
01CC    39                      NOACT:  RTS
```

```
                                        .SBTTL  XFRCTL--ATTACH I/O DRIVER
                                ;THIS ROUTINE DECIDES WHICH I/O DRIVER TO USE GIVEN THE PRIMARY
                                ;ADDRESS IN A. PRIM AND THE DISPATCH TABLE ADDRESS IN THE INDEX REG..
                                ;AFTER I/O, CONTROL IS RETURNED TO CALLER.
                                        .GLOBL  PIATBL
                                ;
01CD  86  000G    XFRCTL:  LDA A  RTRNTG,I          ; TAG IT
01CF  36                   PSH A
01D0  96  000G             LDA A  A.PRIM,D          ; GET DEVICE ADDRESS
01D2  27  08               BEQ     FIL.A    ; IS IT FILE DEVICE
01D4  81  24               CMP A  36.,I
01D6  24  0C               BCC     XFRBNK
01D8  80  1E               SUB A  30.,I     ; IEC ADDRESS?
01DA  2E  01               BGT     1$
01DC  4F                   CLR A           ; ADJUST OFFSET
01DD  4C          1$:      INC A
01DE  48                   ASL A
01DF  BD  0000G   FIL.A:   JSR     JMPX     ; GO OUTPUT BUFFER
01E2  32                   PUL A            ; UNTAG IT
01E3  39                   RTS
01E4  CE  0000G   XFRBNK:  LDX     PIATBL,I ; TRANSFER TO EXTERNAL DEVICE
01E7  6D  00      1$:      TST     0,X
01E9  26  05               BNE     2$       ; TEST IF AT END OF PIATBL
01EB  BD  01C8'            JSR     NIODEV
01EE  31                   INS
01EF  39                   RTS
01F0  A1  05      2$:      CMP A  5,X       ; IF CAN FIND A DEVICE THAT MATCHES
01F2  27  05               BEQ     XFRTBK
01F4  BD  0000G            JSR     A6X
01F7  20  EE               BRA     1$
01F9  96  000G    XFRTBK:  LDA A  BANK,D
01FB  36                   PSH A
01FC  36                   PSH A
01FD  86  000G             LDA A  ITM1TG,I
01FF  36                   PSH A
0200  A6  02               LDA A  2,X
0202  84  3F               AND A  63.,I
0204  BD  0000G            JSR     SETBNK
0207  EE  03               LDX     3,X
0209  AD  06               JSR     6,X
020B  31                   INS
020C  31                   INS
020D  32                   PUL A
020E  84  3F               AND A  63.,I
0210  BD  0000G            JSR     SETBNK
0213  31                   INS
0214  39                   RTS
```

CPSSET--T4051 SET STATUS ADDRESS

```
                                        .SBTTL  CPSSET--T4051 SET STATUS ADDRESS
                                ;THIS ROUTINE SETS THE  MODE DELIMITERS FOR ADDRESS  7,0:
                                ;
                                        .GLOBL  FPEOR,FPEOF,FPNUL,FIXNUM
0215  BD  0000G     CPSSET: JSR     FIXNUM          ; GET NUMBER
0218  96    00G             LDA A   YAXIS,D         ; FIND WHICH ONE TO UPDATE
021A  27    11              BEQ     CPEOL
021C  4A                    DEC A
021D  27    13              BEQ     CPETX
021F  4A                    DEC A
0220  26    16              BNE     CPEX
0222  F7  0000G             STA B   FPNUL
0225  96    00G             LDA A   IOFLGS,D        ; END I/O PROCESSOR CLEANLY
0227  8A    12              ORA A   18.,I
0229  97    00G             STA A   IOFLGS,D
022B  20    08              BRA     CPEX1
022D  F7  0000G     CPEOL:  STA B   FPEOR
0230  20    03             BRA     CPEX1
0232  F7  0000G     CPETX:  STA B   FPEOF
0235  7C  0000G     CPEX1:  INC     YAXIS
0238  39            CPEX:   RTS
```

```
                                        .SBTTL  CPSIEC--IEC CR OR CR/LF FORMAT CONTROLLER
                                        .GLOBL  IECRLF
                                        ;
0239  BD  0000G       CPSIEC: JSR     F1XNUM          ; GET NUMBER
023C  F7  0000G               STA B   IECRLF          ; SAVE IT AS A FLAG
023F  86  12                  LDA A   18.!1           ; END I/O
0241  B7  0000G               STA A   IOFLGS
0244  39                      RTS
```

```
                        .SBTTL  ITTOUT--INTERNAL XFERS. (ERRORS)
                ;
                ;***************************************************************
                ;       INTERNAL XFERS
                ;
                        .IF DF ITT
ITTOUT: LDA A   ERINTR,I        , INTERNAL XFERS. OVER END ARE ILLEGAL
                        STA A   ERRCD,D
                        RTS
                        .ENDC
                        .END
```

SYMBOL TABLE

```
ABRFLG= 0040        AFAIL = 0030        AMAT  = 0010        ARRAY = 0020        ASTR  = 0020
ATLOD = 0008        ATSNTG= XXXXXX G    ATWLD = 0004        A.END = XXXXXX G    A.MAX = XXXXXX G
A.PRIM= XXXXXX G    A.PTR = XXXXXX G    A.SEC = XXXXXX G    A.STAT= XXXXXX G    A.STRT= XXXXXX G
A6X   = XXXXXX G    A8X   = XXXXXX G    BANKTG= XXXXXX G    BANK  = XXXXXX G    BFRSTT= 0020
BLINK = XXXXXX G    BMAT  = 0004        BRKCNT= XXXXXX G    BSTR  = 0008        BUSACT= 0010
CDOPTR= XXXXXX G    CDSPTR= XXXXXX G    CHAR  = XXXXXX G    CHRCNT= XXXXXX G    CLPTR = XXXXXX G
CMAT  = 0001        COLCNT= XXXXXX G    CPEOL   0220R  02   CPETX   0232R  02   CPEX    0238R  02
CPEX1   0236R  02   CPSIEC  0239RG 02   CPSSET  0215RG 02   CPUDEV= 0025        CPUSPC  0091R  02
CPUSTB  0002R  02   CR    = 0000        CRDC3 = 0002        CREOF = 0008        CREOI = 0004
CREOT = 0020        CRETX = 0010        CRLF    0014RG 02   CRLFLF  0012RG 02   CRNORM= 0001
CRSTAT= XXXXXX G    CRVLD = 0080        CSTR  = 0002        CTXN  = XXXXXX G    CURSOR= XXXXXX G
DATDEV= 0022        DEFPNT= XXXXXX G    DIMFLG= 0004        DIRCT = 0080        DISCNT= XXXXXX G
DISSRC= 0008        DL    = XXXXXX G    DP    = XXXXXX G    DREXTA= XXXXXX G    DREXTB= XXXXXX G
DSDRAW= XXXXXX G    DSFONT= XXXXXX G    DSFULL= XXXXXX G    DSMOVE= XXXXXX G    DSPDEV= 0020
DSPOUT= XXXXXX G    DSPSPC  0096R  02   DSPSTB  0088R  02   DT    = XXXXXX G    EDTBFR= XXXXXX G
ENDKEY= 0040        EOFTYP= 0038        EOLCHR= XXXXXX G    EOLTG = XXXXXX G    EOSTG = XXXXXX G
ERRTSN= XXXXXX G    ERRTRA= XXXXXX G    ERRTRB= XXXXXX G    ERRDMN= XXXXXX G    ERROM = XXXXXX G
ERFBFR= XXXXXX G    ERFILE= XXXXXX G    ERIOE = XXXXXX G    ERNIOD= XXXXXX G    ERNSEP= XXXXXX G
ERRCD = XXXXXX G    ERTERM= XXXXXX G    ERUNDF= XXXXXX G    ESTG  = XXXXXX G    EXIT    010FR  02
EXQ     00E1R  02   EXQ.2   011ER  02   EXTFLG= 0080        FILDEV= 0000        FILEOT= XXXXXX G
FIL.A   010FR  02   FIXNUM= XXXXXX G    FIX1A = 0003        FIX1B = 0004        FMTPNT= XXXXXX G
FMTVLD= 0008        FNFLG = 0010        FORTG = XXXXXX G    FPNASC= XXXXXX G    GLBFLG= XXXXXX G
COSTG = XXXXXX G    IECOUT= XXXXXX G    IECRLF= XXXXXX G    INXTG = XXXXXX G    INPUTK= 0001
IOBFR1= XXXXXX G    IOFLGS= XXXXXX G    IOFUNC= XXXXXX G    ISL   = XXXXXX G    ISP   = XXXXXX G
ITM1TG= XXXXXX G    ITM2TG= XXXXXX G    ITTDEV= 0024        JMPX  = XXXXXX G    KBDEV = 001F
KBFLAG= XXXXXX G    KBIN  = XXXXXX G    KEYFLG= 0010        KEYSTK= XXXXXX G    LASTYP= 0020
LBRKTG= XXXXXX G    LCLFLG= XXXXXX G    LDAX  = XXXXXX G    LDXX  = XXXXXX G    LENGTH= XXXXXX G
LISTTG= XXXXXX G    LNNOTG= XXXXXX G    LOCTG = XXXXXX G    LSP   = XXXXXX G    LSTFMT= 0002
MTBFR = XXXXXX G    MTEIND= XXXXXX G    MTKILL= XXXXXX G    MTMARK= XXXXXX G    MTPDEV= 0021
MTPD2 = 0023        MTPOUT= XXXXXX G    MTPSPC  0098R  02   MTPSTB  00C5R  02   MTSSET= XXXXXX G
NIODEV  01C8RG 02   NLPTR = XXXXXX G    NOACT   U1CCR  02   NOKEY = 0080        NOOUT = XXXXXX G
NOWRIT= 0001        NSCRM   0266R  02   NTAPTR= 0008        NTATTR= 0004        NTDIMS= 0009
NTDLEN= 0005        NTLINK= 0000        NTNAME= 0002        NTPTR = XXXXXX G    NTRELY= 0010
NTSPTR= 0008        NTVAL = 0005        NTWCOL= 0007        NTWLEN= 0007        NTWROW= 0005
NULLTG= XXXXXX G    OBJATR= 0002        OBJBCK= 0003        OBJDT = 0005        OBJLEN= 0000
OKDONE  015ER  02   OKK     0176R  02   OKKK    0190R  02   OKLOOP  0134R  02   OKSND   0121R  02
OKSNOT  0169R  02   OKSVIT  014DR  02   ONSFLG= 0002        OPRADR= XXXXXX G    OUTBFR  0110RG 02
OUTLIN  0058R  02   OUTTBL  0188R  02   PARTG = XXXXXX G    PARM  = 0008        PGMATR= 0002
PGMBP = 0005        PGMCD = 0009        PGMFP = 0003        PGMLEN= 0000        PGMLNK= 0007
PGMPTR= XXXXXX G    PGMTG = XXXXXX G    PIATBL= XXXXXX G    PLOSTG= XXXXXX G    PNDEOF= XXXXXX G
PNTFLG= XXXXXX G    PNTNTG= XXXXXX G    PNTSTG= XXXXXX G    POINT = XXXXXX G    PPEOF = XXXXXX G
PPEOR = XXXXXX G    PPMODE= XXXXXX G    PPNUL = XXXXXX G    PRIDEF= 0001        PRIDFY  0024RG 02
PRISPC  0069R  02   PRISTG  0000RG 02   PRIVAL   001ARG 02   PRTTG = XXXXXX G    PSCTG = XXXXXX G
PSHFPN= XXXXXX G    PUTBT   0009R  02   PUTBYT  0009RG 02   RECLEG= 0004        REFIL = 00C0
RMAT  = 0040        RPTCYL= 0080        RSTR  = 0080        RTRNTG= XXXXXX G    RUNFLG= 0080
RUNN  = 0002        R0    = XXXXXX G    R1    = XXXXXX G    R10   = XXXXXX G    R11   = XXXXXX G
R12   = XXXXXX G    R13   = XXXXXX G    R14   = XXXXXX G    R15   = XXXXXX G    R16   = XXXXXX G
R17   = XXXXXX G    R18   = XXXXXX G    R19   = XXXXXX G    R2    = XXXXXX G    R20   = XXXXXX G
R21   = XXXXXX G    R22   = XXXXXX G    R23   = XXXXXX G    R3    = XXXXXX G    R4    = XXXXXX G
R5    = XXXXXX G    R6    = XXXXXX G    R7    = XXXXXX G    R8    = XXXXXX G    R9    = XXXXXX G
SBP   = XXXXXX G    SCALER= 0040        SCMSED= XXXXXX G    SCM1    0260R  02   SCRM    024FR  02
SCRME   0262R  02   SCRMO   0269RG 02   SCRMO.  0268R  02   SCRM$   0245RG 02   SCRTCH= XXXXXX G
SECDEF= 0002        SEMIC = 0002        SEMITG= XXXXXX G    SETBNK= XXXXXX G    SNDBFR  01A0RG 02
SNDIT = 0080        SPACE = 0020        SPALOP  004ER  02   SPCDFT  008BR  02   SPCFND  00A8R  02
SPCNOT  0079R  02   SPCSCN  009ER  02   SP.OUT  005DR  02   STAT37= XXXXXX G    STPFLG= 0040
STPKEY= 0020        STRING= 0010        SYSERR= XXXXXX G    TAB     0033R  02   TABCNT= XXXXXX G
TABLOP  0144AR 02   TABPTR= XXXXXX G    TAGPTR= XXXXXX G    TB.OUT  0067R  02   TCOL  = XXXXXX G
```

```
OUTCTL OUTPUT CONTROL ROUTINES        RT-11 MMAC  VM02-10   14-OCT-76 01:37:27 PAGE 16+
SYMBOL TABLE

TRCFLG=  0020        T510EV=  0025        UNDEF =  0080        VALTG = XXXXXX G     VALUND=  0040
WREAD    0268R   02 XAXIS = XXXXXX G      XERBNK   01E4R   02  XERCTL   01CDRG  02  XERTBK   01F9R   02
YAXIS = XXXXXX G
. ABS.   0000    00
         0001    01
OUTCTL   0279    02
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2210. WORDS
.SY:OUTCTL/C<DK1:SELCL1.OUTCTL
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
IMXTG   1-129#
INPUTK  1-219#
IOBFR1  1-266#
IOELGS  1-246#  5-18    12-13   12-15*  13-7*
IOFUNC  1-190#  8-88    8-94    14-13
ISL     1-290#
ISP     1-290#
ITM1TG  1-136#  11-34
ITM2TG  1-137#
ITT     16-6
ITTDEV  1-260#
JMPX    1-55#   11-18
KBDEV   1-255#
KBFLAG  1-210#
KBIN    1-208#
KEYFLG  1-35#
KEYSTK  1-52#
LASTYP  5-13#   5-21
LBRKTG  1-139#
LCLFLG  1-24#
LDAX    1-65#
LDXX    1-60#
LENGTH  1-240#  2-8
LISTTG  1-127#
LNNOTG  1-135#
LOCTG   1-145#
LSP     1-46#
LSTFMT  1-227#  8-56
MTBFR   1-267#
MTFIND  7-20    7-20#
MTKILL  7-18    7-18#
MTMARK  7-19    7-19#
MTPD2   1-259#
MTPDEV  1-257#  6-21
MTPOUT  10-4#   10-11
MTPSPC  6-22    6-42#
MTPSTB  6-42    7-18#
MTSSET  7-21    7-21#
NIODEV  10-4#   10-7    10-10   10-17#  11-24
NLPTR   1-21#
NORACT  10-9    10-19#
NOKEY   1-209#
NOOUT   1-225#  8-44    9-17    9-19*
NOURIT  1-226#  8-45
NSCRM   14-12   14-15   14-29#
NTAPTR  1-93#
NTATTR  1-80#
NTDIMS  1-92#
NTDLEN  1-95#
NTLTHK  1-77#
NTNAME  1-78#
NTPTR   1-22#
NTRELV  1-215#
NTSPTR  1-97#
NTVAL   1-87#
NTWCOL  1-91#
NTWLEN  1-96#
```

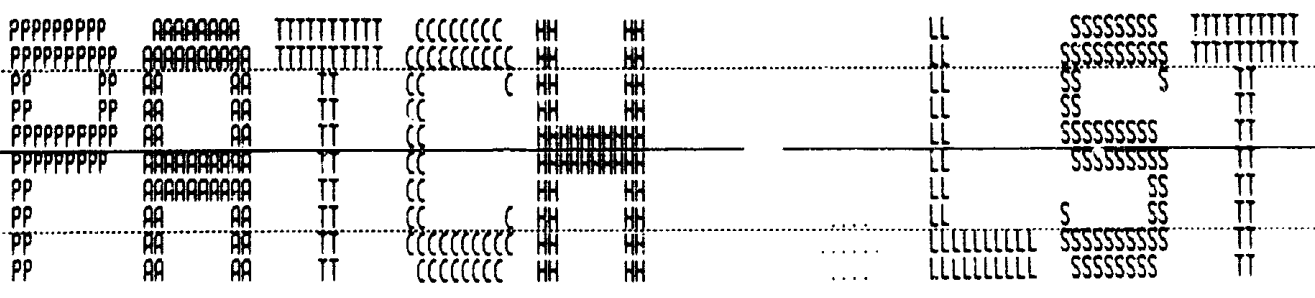
CROSS REFERENCE TABLE (CREF V01-03)

```
NTMROW  1-90#
NULLTG  1-122#
OBJATR  1-102#
OBJBCK  1-103#
OBJDT   1-104#
OBJLEN  1-101#
OKDONE  8-79    8-83#
OKK     8-90    8-92    8-94#
OKKK    8-96    8-100   8-102   8-107#
OKLOOP  8-61#   8-82
OKSND   8-46    8-52#
OKSNDT  8-57    8-88#
OKSWIT  8-64    8-66    8-74#
ONSFLG  1-29#
OPRADR  1-238#
OUTBFR  1-278#  5-30    8-22    8-41#   9-20
OUTLIN  5-36    5-46#
OUTTBL  8-107   10-5#
PAETG   1-133#
PARM    1-85#
PGMATR  1-109#
PGMBP   1-111#
PGMCD   1-113#
PGMFP   1-110#
PGMLEN  1-108#
PGMLNN  1-112#
PGMPTR  1-48#
PGMTG   1-128#
PIATBL  11-5#   11-21
PLOSTG  1-123#
PNDEOF  1-45#
PNDFLG  1-44#
PNTNTG  1-132#
PNTSTG  1-130#
POINT   1-241#  2-6     6-29
PPEOF   12-4#   12-19%
PPEOR   12-4#   12-17%
PPMODE  1-224#
PPNUL   12-4#   12-12%
PRIDEF  1-206#
PRIOFT  5-9#    5-16#   6-55
PRISPC  2-12    4-16    6-17#
PRISTG  1-279#  2-4#
PRIVAL  1-282#  4-12#
PRTTG   1-141#
PSCTG   1-131#
PSHFPN  1-285#  6-30
PUTBT.  8-13#
PUTBYT  1-287#  3-7     5-42    5-51    8-125
R0      1-38#
R1      1-38#
R10     1-38#
R11     1-38#
R12     1-38#
R13     1-38#
R14     1-38#
R15     1-38#
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-3#
SPC 7-7# 7-13 7-14 7-15 7-16 7-18 7-19 7-20 7-21 7-23 7-24



```
                                        .TITLE  PATCH   FIXES TO THE 4051 FIRMWARE.
                                        .IDENT  /JG0005/
                                ;
                                ;       THIS FITS IN THE SPACE FROM X'8746' AND X'8780'
                                ;
                0000                    .ASECT
                8746                    . = X8746
                                ;
                                ;       PASTCH CODE FOR PROBLEM 16.
                                ;
                                        .GLOBL  BUG16A,BUG16B,BUG16C
8746    08              BUG16A: INX
8747    08                      INX
8748    BD      0000G           JSR     BUG16B
8748    09                      DEX
874C    09                      DEX
874D    A6      01              LDA A   1,X
874F    7E      0000G           JMP     BUG16C
                                ;
                                ;       PATCH FOR BUG 7.
                                ;
                                        .GLOBL  IOPAT1,IOSNTP,DB
8752    FF      0000G   IOPAT1: STX     DB              ;FOR PAGE FULL
8755    96      00G             LDA A   IOSNTP,D        ;CLSO IN NAME TABLE
8757    8A      80              ORA A   128.,I
8759    39                      RTS
                                ;
                                ;
                                ;       THIS PATCH TO FIX CLOSE ( BUG #24 )
                                ;
                                        .GLOBL  CLOSP1, MTCLOS, CRTRST, ERRCD, HALTR
875A    BD      0000G   CLOSP1: JSR     MTCLOS          , CLOSE THE MAGTAPE
875D    BD      0000G           JSR     CRTRST
8760    7F      0000G           CLR     ERRCD
8763    7E      0000G           JMP     HALTR           ; EXIT!!!!
                                ;
                                ;
                                ;
                                ;       FIX FOR BUG 22 THIS FIX WILL NOT WORK UNTIL EVLEN IS PATCHED
                                ;       ALSO
                                ;
                                        .GLOBL  FIX22,EOLTG,LNEVL
8766    86      00G     FIX22:  LDA A   EOLTG,I
8768    36                      PSH A
8769    7E      0006G           JMP     LNEVL+6         ;SKIP CLR A STAT
                                ;
                                ;
                                ;       FIX FOR IEC COPY MODE USING WBYTE  A: FORMAT
                                ;
                                        .GLOBL  WBYP3, PPMODE, INTACP
                                        .GLOBL  PIASRQ, UNADR2
```

```
876C  BD  0000G   WBYP3:  JSR    INTACP          ; INITIALIZE THE BUS
876F  86  0C              LDA A  H0C,I  ; SET UP TO CLEAR ATN ETC
8771  D6  00G             LDA B  PPMODE,D        ; SEE IF DOING WBY  N: AND NEED OFF BUS
8773  27  02              BEQ    1$
8775  86  9C              LDA A  H9C,I  ; ALL LINES EXCEPT "REN" ARE SET HIGH
8777  B7  0000G   1$:     STA A  PIASRQ
877A  31                  INS                    ; CLEAN OF BAD RETURN ADDRESS
877B  31                  INS
877C  7E  0000G           JMP    UNADR2          ; RETURN TO FINISH CLEAN UP
                  ;
                  ;
      0001'               .END
```

```
PATCH  FIXES TO THE 4051 FIRMW RT-11 MMAC  VM02-10   14-OCT-76 01:37:48 PAGE 1+
SYMBOL TABLE

BUG16A    8746 G      BUG16B= XXXXXX G      BUG16C= XXXXXX G      CLOSM1    875A G      CRTRST= XXXXXX G
DB     = XXXXXX G     EOLTG = XXXXXX G      ERRCD = XXXXXX G      FIX22     8766 G      HALTR = XXXXXX G
INTACP= XXXXXX G      IOPAT1    8752 G      IOSNTP= XXXXXX G      LNEVL = XXXXXX G      MTCLOS= XXXXXX G
PIASRQ= XXXXXX G      PPMODE= XXXXXX G      UNADR2= XXXXXX G      WBYP3     876C G
  ABS.    872E    00
          0000    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3415. WORDS
.SY:PATCH/C<DK1:SEICLI.PATCH
```

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-38

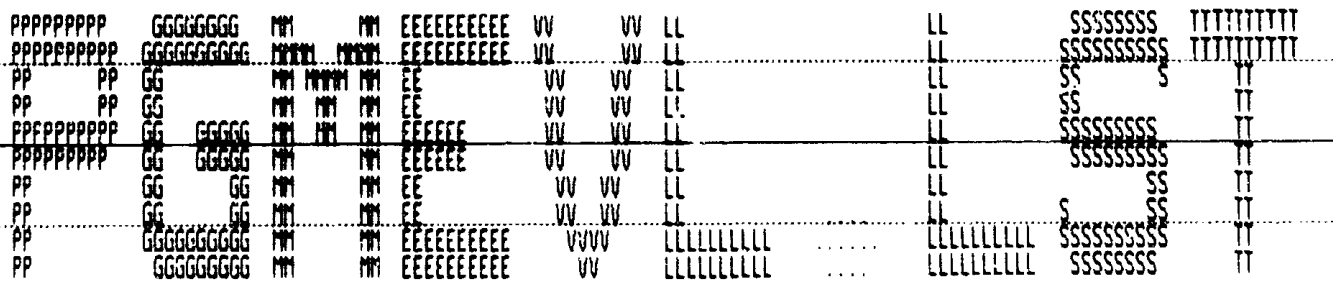

PPPPPPPP GGGGGGGG MM MM EEEEEEEEEE VV VV LL LL SSSSSSSS TTTTTTTTTT
PPPPPPPPP GGGGGGGGGG MMMM MMMM EEEEEEEEEE VV VV LL LL SSSSSSSSSS TTTTTTTTTT
PP PP GG MM MMM MM EE VV VV LL LL SS S TT
PP PP GG MM MM MM EE VV VV LL LL SS TT
PPPPPPPPP GG GGGGG MM MM MM EEEEEE VV VV LL LL SSSSSSSSS TT
PPPPPPPP GG GGGGG MM MM EEEEEE VV VV LL LL SSSSSSSS TT
PP GG GG MM MM EE VV VV LL LL SS TT
PP GG GG MM MM EE VV VV LL LL S SS TT
PP GGGGGGGGGG MM MM EEEEEEEEEE VVVV LLLLLLLLLL LLLLLLLLLL SSSSSSSSS TT
PP GGGGGGGG MM MM EEEEEEEEEE VV LLLLLLLLLL LLLLLLLLLL SSSSSSSS TT

TABLE OF CONTENTS

```
                                        .TITLE  PGMEVL  LINE EVALUATION ROUTINES
                                        .IDENT  /BE10E1/
                                        ;
                                        .GLOBL  LNEVL,PGMEVL
                                        .GLOBL  COMPR,TYPARG,SETARG,TYPRES,LOCTG,SYSERR
                                        .GLOBL  TSTINT,IDLE,END,QTRNSL,QUSRFN,OPRCL
                                        .GLOBL  SNGMAT,LITREL,SCLMAT,MATSCL,MATMAT
                                        .GLOBL  OCTBL,CMDTBL,OPRTBL
                                        .GLOBL  BRKCOD
                                        .GLOBL  IOPROC,FILES,GRAPH,DSPLY,TAPE,TAPFIL
                                        .GLOBL  FUDGH,FUDGL,SLOPH,SLOPL,SIZERR
                                        .GLOBL  INRGTE,R10,R16,DSPCHR
                                        .GLOBL  BRKCNT,A.STAT,MSKCTR
                                        .GLOBL  XEQSTK,XEQSP
                                        .GLOBL  AOX
                                        ;
                                        ;       INPUTS
                                        ;               CONTROLS SHOULD BE SET UP.
                                        ;               NLPTR SHOULD HAVE ADDR OF NEXT LINE.
                                        ;               CLPTR SHOULD HAVE ADDR OF CURRENT LINE
                                        ;               LCLFLG & GLBFLG BITS WILL CONTROL FLOW OF CONTROL
                                        ;
                                        ;       OUTPUTS
                                        ;               CALLS TO OPERATOR AND SYSTEM ROUTINES
                                        ;
                                        ;       NOTES
                                        ;               IF NLPTR IS NOT ZERO IT WILL BECOME CLPTR.
                                        ;               IF NLPTR IS ZERO CLPTR IS USED TO CHAIN TO NEXT LINE
                                        ;               IF BOTH NLPTR AND CLPTR ARE ZERO EXECUTION STOPS
                                        ;
                                        ;
                                        ;               THE RUN BIT IS RESET IN THE IDLE LOOP
                                        ;
                                        .SBTTL  XXX PGMEVL      PROGRAM EVALUATION ROUTINE
                                        ;
0000    81      00G     LNSIZE: CMP A   SIZERR,I        ;IS IT SIZE ERROR
0002    22      12              BHI     LNEXT           ;IF NOT STOP NOW
0004    7F      0000G           CLR     ERRCD           ;I WANT SIZE ERROR TO RAISE ON UNIT ONLY
0007    D6      00G             LDA B   ERRCDB,D        ;HAS ANOTHER SIZE ERROR OCCURED
0009    26      08              BNE     LNSZOK          ;IF SO FORGET NEW ONE
000B    97      00G             STA A   ERRCDB,D        ;SET SIZE ERROR IN ERRPR CODE BACKUP AREA
000D    86      40              LDA A   64.,I           ;NOW RAISE SIZE ERROR
000F    9A      00G             ORA A   PNDFLG,D        ;IT IS PENDING INTERRUPT NOW
0011    97      00G             STA A   PNDFLG,D
0013    7E      0087    LNSZOK: JMP     OBJGO           ;CONTINUE IN LINE
                                        ;
0016    86      00G     LNEXT:  LDA A   EQLTG,I         ;CLEAR TO END OF STACK
0018    9F      00E             STS     R0,D            ;START AT END OF STACK
001A    BD      0000G           JSR     LOCTG           ;CALL LOC TAG ROUTINE
001D    9E      00G             LDS     R0,D            ;RESET STACK PTR
001F    96      00G             LDA A   ERRCD,D         ;IS ERROR CODE SET
0021    26      21              BNE     PGMHLT
0023    86      80              LDA A   EXTFLG,I        ;EXIT REQ
0025    95      00G             BIT A   LCLFLG,D
0027    26      2E              BNE     PGMEXT
0029    96      00G             LDA A   GLBFLG,D        ;IN RUN MODE
002B    2A      17              BPL     PGMHLT
```

```
                                ;
                                ;CLEAR LOCAL FLAGS
                                ;TEST INTERRUPT PENDING
                                ;SEE IF ERROR CODE BYTE GOT SET
                                ;
                                ;STILL IN RUN MODE
                                ;
                                ;CLEAR LCLFLG
                                ;SEE IF IDLE ADDR ON STACK
                                ;IF FROM IDLE LOOP I CAN RETURN
                                ;IF USER FN ACTIVE EXIT
                                ;BASIC PROGRAM UNDOES RECURSIONS
                                ;
                                ;RETURN TO CALLER
                                ;
                                ;DO AN IMPLICIT END
                                ;
                                ;SET RUN MODE FOR PROGRAM
                                ;
                                ;TAG STACK FOR END OF LINE
                                ;RESET I/O SYSTEM
                                ;IS IT USER FUNCTION
                                ;IS IT IMMEDIATE EXECUTE LINE
                                ;MOVE TO NEXT LINE
                                ;IF ZERO TRY CL*TR AND ADVANCE
                                ;IS CLPTR DEFINED
                                ;I THINK IT IS TIME TO STOP
                                ;TRY NEXT LINE PTR FROM OBJ STR
                                ;END OF PGM - TRA
                                ;HAVE CURRENT LINE PTR NOW
                                ;NOW SET UP NLPTR
                                ;SET UP PTR TO FIRST TOKEN
                                ;
                                ;CLEAR LOCAL FLAG BITS
                                ;
                                ;TEST FOR MIN STACK SPACE
                                ;CALC LSF+FUDGE-SP
```

```
 115  0098  C6  000        LDA B  FUDGL,I
 116  0090  D8  01G        ADD B  LSP+1,D
 117  009F  99  000        ADC A  LSP,D
 118  00A1  D0  01G        SUB B  R0+1,D
 119  00A3  92  000        SBC A  R0,D
 120  00A5  2A  7C         BPL    LNWSFL          ;IF RECOVER COUNT IS POS I CAN'T RUN
 121  00A7  96  000  LNSOK: LDA A  GLBFLG,D        ;IN TRACE MODE
 122  00A9  85  20         BIT A  TRCFLG,I
 123  00AB  27  03         BEQ    OBJLP
 124  00AD  7E  0220'      JMP    LNTRC
 125  00B0  96  000  OBJLP: LDA A  ERRCD,D         ;WILL STOP IF ERROR CODE NOT ZERO
 126  00B2  27  03         BEQ    OBJGO           ;IF NOT SET I CAN RUN LINE
 127  00B4  7E  0000'      JMP    LNSIZE
 128  00B7  86  40   OBJGO: LDA A  GBRFLG,I
 129  00B9  95  000        BIT A  LCLFLG,D         ;TEST TO SEE IF I SHOULD STOP
 130  00BB  27  03         BEQ    OBJGOB
 131  00BD  7E  0016'      JMP    LNEXT
 132  00C0  CE  0080' OBJGOB: LDX    OBJLP,I         ;SET UP POSSIBLE RETURN POINT
 133  00C3  DF  01G        STX    DREXTA+1,D
 134  00C5  CE  0002G      LDX    XEQSTK+2,I      ;SET STACK POINTER TO ORG OF STAK
 135  00C8  DF  000        STX    XEQSP,D
 136  00CA  DE  00G        LDX    NTPTR,D         ;GET NEXT TOKEN
 137  00CC  A6  00         LDA A  0,X
 138  00CE  97  000        STA A  CTKN,D
 139  00D0  08             INX                    ;BUMP PTR
 140  00D1  DF  00G        STX    NTPTR,D
 141  00D3  81  00G        CMP A  LSTCOD,I        ;IS IT VALID
 142  00D5  22  68         BHI    OBJFL           ;IF TOO BIG I AM LOST
 143  00D7  81  00G        CMP A  FNACOD,I        ;DELETE SPACE FROM FUNCTION CODES
 144  00D9  23  08         BLS    OBJNFN          ;NOT FUNCTION
 145  00DB  80  19         SUB A  25.,I           ;BIAS REST OF TOKENS
 146  00DD  81  00G        CMP A  FNACOD,I        ;IN FN RANGE
 147  00DF  22  02         BHI    OBJNFN          ;NO
 148  00E1  86  00G        LDA A  FNACOD,I        ;ALL FUNCTIONS ARE FNA NOW
 149  00E3  97  00G  OBJNFN: STA A  OPRADR,D       ;SAVE TOKEN FOR NOW
 150  00E5  44             LSR A                  ;GET DISP TO DISPATCH CODE
 151  00E6  CE  0000G      LDX    OCTBL,I         ;OPERATOR CONTROL TABLE BASE
 152  00E9  8D  0000G      JSR    LDAX            ;GET OBJECT BYTE
 153  00EC  25  04         BCS    OBJLOW          ;IF NO CARRY I NEED HIGH HALF OF A
 154  00EE  54             LSR B                  ;SHIFT HIGH HALF DOWN
11111
 155  00EF  54             LSR B
 156  00F0  54             LSR B
 157  00F1  54             LSR B
 158  00F2  C4  0F   OBJLOW: AND B  15.,I          ;CLEAN UP REG
 159  00F4  96  00G        LDA A  OPRADR,D         ;GET TOKEN
 160  00F6  80  00G        SUB A  BRKCOD,I         ;IS IT FUNCTION OR COMMAND
 161  00F8  25  0A         BCS    OBJOPR          ;IN COMMAND CLASS
 162                       ;
 163  00FA  CB  0F         ADD B  15.,I           ;BIAS COMMAND DISPATCH
 164  00FC  CE  0000G      LDX    CMDTBL,I        ;GET BASE
 165  00FF  8D  0F         BSR    OBJCOM          ;GO JOIN COMMON CODE
 166  0101  7E  0080'      JMP    OBJLP           ;LOOP NOW
 167                       ;
 168  0104  D7  01G  OBJOPR: STA B  OPRADR+1,D      ;SAVE CLASS TEST DISPATCH ADDR
 169  0106  8D  0000G      JSR    SETGRG
 170  0109  96  00G        LDA A  OPRADR,D         ;GET TOKAN AGAIN
 171  010B  D6  01G        LDA B  OPRADR+1,D       ;AND CLASS TEST ALSO
```

```
172          0100  CE  0000G                 LDX   OPRTBL,I          ;GET ADDR OF ROUTINE TABLE
173
174          0110  48                  OBJCOM: ASL A                 ;DOUBLE TABLE INDEX
175          0111  BD  0000G                   JSR   LDXX              ;GET ADDR OF SERVICE ROUTINE
176          0114  DF    00G                   STX   OPRADR,D          ;SAVE IT
177          0116  CE  0145'                   LDX   CLASS,I           ;GET ADD OF CLASS TEST/SETUP TABLE
178          0119  17                        TBA
179          011A  BD  0000G                   JSR   LDXX
180          011D  CE  015C'                   LDX   CLASST,I
181          0120  7E  0000G                   JMP   JMPAX
182
183          0123  7D  0000G           LNWSFL: TST   GLBFLG            ;IF IN CALC MODE I WILL START WITH
184          0126  2B    08                    BMI   LNWCMP            ;   SLOP BYTES OF SPACE
185          0128  C0    00G                   SUB B SLOPL,I
186          012A  82    00G                   SBC A SLOPH,I
187          012C  2A    02                    BPL   LNWCMP            ;IF RECOVERY SPACE IS STILL POS SKIP
188          012E  4F                          CLR A
189          012F  5F                          CLR B
190          0130  37                  LNWCMP: PSH B
191          0131  36                          PSH A
192          0132  96    00G                   LDA A ITM1TG,I
193          0134  36                          PSH A
194          0135  BD  0000G                   JSR   COMPR
195          0138  96    00G                   LDA A ERRCD,D           ;DID IT WORK
196          013A  26    03                    BNE   LNWFER
197          013C  7E  0042'                   JMP   LNSOK
198
199          013F  7E  0016'           LNWFER: JMP   LNEXT             ;KILL EXECUTION
200
201          0142  BD  0000G           OBJFL:  JSR   SYSERR            ;I AM LOST
202
203          0145  68                  CLASS:  BYTE  CTCALL-CLASST     ;CALL ONLY JMP POINT
204                                             .IRPC Q,123456789ABCDE
205                                             .BYTE CLASS'Q-CLASST
206                                             .ENDM
207          0154  66                          .BYTE CTPULL-CLASST     ;CALL ONLY ENTRY POINT
208                                             .IRPC Q,IFBDTLP
209                                             .BYTE SETUP'Q-CLASST
210                                             .ENDM
```

```
                                        .SBTTL  XXX CLASSX      OPERAND CLASS TESTS
                                ;
015C                    CLASST:                         ;DISPATCH JUMP OFF POINT
                                ;
                                ;       CLASS 3 - IF
                                ;
015C    BD      0000G   CLASS3: JSR     TYPARG
                                ;
                                ;       CLASS 1 - FOR, GOTO, GOSUB, AT, LB, PER, FNX, FNASGN, TYP
                                ;
015F    96      00G     CLASS1: LDA A   R4,D
0161    27      61              BEQ     CTCALL
0163    20      52              BRA     CTERR
                                ;
                                ;       CLASS 2 - NOT, UNARY MINUS, UNARY PLUS
                                ;                 SIN, COS, TAN, ATN, ACS, ASN, EXP, LOG, LGN, INT,
                                ;                 RND, SIG, SQR, ABS
                                ;
0165    96      00G     CLASS2: LDA A   R4,D
0167    27      5B              BEQ     CTCALL
0169    BD      0000G           JSR     TYPRES
016C    81      50              CMP A   RMAT+AMAT,I
016E    27      5A              BEQ     CTSNGM
0170    20      45              BRA     CTERR
                                ;
                                ;       CLASS 4 - MAX, MIN, AND, OR, MLT, DIV, UP, PLUS, MINUS
                                ;
0172    BD      0000G   CLASS4: JSR     TYPARG
0175    27      40              BEQ     CTCALL
0177    20      09              BRA     CLS5A
                                ;
                                ;       CLASS 5 - =, (), (=, (, ), )=
                                ;
0179    BD      0000G   CLASS5: JSR     TYPARG
017C    27      46              BEQ     CTCALL
017E    81      28              CMP A   ASTR+BSTR,I
0180    27      48              BEQ     CTLR
0182    BD      0000G   CLS5A    JSR     TYPRES
0185    81      50              CMP A   RMAT+AMAT,I
0187    27      4A              BEQ     CTMS
0189    81      44              CMP A   RMAT+BMAT,I
018B    27      43              BEQ     CTSM
018D    81      54              CMP A   RMAT+AMAT+BMAT,I
018F    27      45              BEQ     CTMM
0191    20      24              BRA     CTERR
                                ;
                                ;       CLASS 6 - LEN, ASC, VAL
                                ;
0193    96      00G     CLASS6: LDA A   R4,D
0195    81      20              CMP A   ASTR,I
0197    27      28              BEQ     CTCALL
]]]]]
0199    20      1C              BRA     CTERR
                                ;
                                ;       CLASS 7 - CHR, STR
                                ;
019B    BD      0000G   CLASS7: JSR     TYPRES
019E    81      80              CMP A   RSTR,I
```

```
                                                BEQ     CTCALL
                                                BRA     CTERR
                        ;
                        ;       CLASS 8 - CAT
                        ;
                        CLASS8: JSR     TYPARG
                                JSR     TYPRES
                                CMP A   RSTR+ASTR+BSTR,I
                                BEQ     CTCALL
                                BRA     CTERR
                        ;
                        ;       CLASS 9 - TRN, INV
                        ;
                        CLASS9: JSR     TYPRES
                                CMP A   RMAT+AMAT,I
                                BEQ     CTCALL
                        ;
                        ;       COMMON CLASS TEST EXIT POINTS
                        ;
                        ;       CTERR - AN ERROR HAS OCCURED
                        ;
                        CTERR:  LDA A   ERRCD,D         ;WAS IT SET BY TYPRES
                                BNE     CTKILL          ;DON'T CHANGE ERROR CODE
                                LDA A   ERDOMN,I        ;SET ERROR CODE
                                STA A   ERRCD,D
                        CTKILL: JMP     LNEXT           ;MUST STOP NOW
                        ;
                        ;       CTCALL - CALL TOKEN EXEC ROUTINE
                        ;
                        CTPULL: PUL A
                                PUL A
                        CTCALL: JSR     OPRCL           ;USE GENERAL PURPOSE CALL
                                JMP     08JLP           ;GO TRY NEXT TOKEN
                        ;
                        ;       CTSNGM - CALL SINGLE MATRIX ROUTINE
                        ;
                        CTSNGM: JMP     SNGMAT          ;CALL ROUTINE FOR ALL OPERATORS
                        ;
                        ;       CTLR - LITERAL RELATION ROUTINES
                        ;
                        CTLR:   JMP     LITREL
                        ;
                        ;       CTSM - SCALER MATRIX ROUTINE
                        ;
                        CTSM:   JMP     SCLMAT          ;ROUTINE TO CALL
                        ;
                        ;       CTMS - MATRIX SCALER ROUTINE
                        ;
                        CTMS:   JMP     MATSCL
                        ;
                        ;       CTMM - MATRIX MATRIX ROUTINE
                        ;
                        CTMM:   JMP     MATMAT
                        ;
                        ;       CLASS 10 - DRAW, RDRAW, MOVE, RMOVE
                        ;
                        CLASSA: JSR     TYPARG
```

```
                                ;
                                ;       CLASS B - SEG. REP
                                ;
01E4  BD  0000G   CLASSB: JSR   TYPARG
01E7  BD  0000G           JSR   TYPARG
01EA  BD  0000G           JSR   TYPRES
01ED  81  0D              CMP A RSTR+0STR,I
01EF  27  D3              BEQ   CTCALL
01F1  20  C4              BRA   CTERR
                                ;
                                ;       CLASS C - POS
                                ;
01F3  BD  0000G   CLASSC: JSR   TYPARG
01F6  BD  0000G           JSR   TYPARG
01F9  81  28              CMP A ASTR+0STR,I
01FB  27  C7              BEQ   CTCALL
01FD  20  B8              BRA   CTERR
                                ;
                                ;       CLASS D - SUM
                                ;
01FF  96  00G     CLASSD: LDA A R4,0
0201  81  10              CMP A 0MAT,I
0203  27  BF              BEQ   CTCALL
0205  20  B0              BRA   CTERR
                                ;
                                ;       CLASS E - MPY
                                ;
0207  BD  0000G   CLASSE: JSR   TYPARG
020A  BD  0000G           JSR   TYPRES
020D  81  54              CMP A RMAT+0MAT+0MAT,I
020F  27  B3              BEQ   CTCALL
0211  20  A4              BRA   CTERR
```

```
                                        .SBTTL  XXX SETUPX      SYSTEM SETUP FUNCTIONS
                                ;
                                ;       SETUPI - IOPROC CALL
                                ;
0213    7E      0000G   SETUPI:  JMP     IOPROC
                                ;
                                ;       SETUPF - FILE SET UP
                                ;
0216    7E      0000G   SETUPF:  JMP     FILES
                                ;
                                ;       SETUPD - DISPLAY SET UP
                                ;
0219    7E      0000G   SETUPD:  JMP     DSPLY
                                ;
                                ;       SETUPT - MAG TAPE
                                ;
021C    7E      0000G   SETUPT:  JMP     TAPE
                                ;
                                ;       SETUPL - LIST COMMANDS
                                ;
021F    7E      0000G   SETUPL:  JMP     TAPFIL
                                ;
                                ;       INC BRACKET COUNT AND PUSH
                                ;
0222    7C      0000G   SETUPB:  INC     BRKCNT
                                ;
                                ;       PUSH TAG ON STACK
                                ;
0225    32              SETUPP:  PUL A
0226    32                       PUL A
0227    96      01G              LDA A   OFRADR+1,D
0229    36                       PSH A
022A    7E      0080'            JMP     OBJLP
```

```
                                        .SBTTL  XXX LNTRC       LINE TRACE ROUTINE
                                        ;
0220    96      00G     LNTRC:  LDA A   LCLFLG,D        ;NO TRACE ON DEF FN LINES
022F    85      10              BIT A   FNFLG,I
0231    26      1F              BNE     TRCEXT
0233    DE      00G             LDX     CLPTR,D         ;GET LINE NUMBER
0235    EE      07              LDX     PGMLNN,X        ;GET LINE NUMBER
0237    27      19              BEQ     TRCEXT          ;IF NO LINE NUMBER FORGET IT
0239    BD      0000G           JSR     INAGTE          ;CONVERT TO ASCII
023C    CE      FFFFG           LDX     R16-1,I
023F    86      23              LDA A   '#,I
0241    DF      00G     TRCLP:  STX     R10,D
0243    BD      0000G           JSR     DSPCHR
0246    DE      00G             LDX     R10,D
0248    08                      INX
0249    A6      00              LDA A   0,X
024B    26      F4              BNE     TRCLP
024D    86      0D              LDA A   13.,I           ;CR CODE
024F    BD      0000G           JSR     DSPCHR
0252    7E      00B0'   TRCEXT: JMP     @BJLP
                                ;
                0001'           .END
```

```
PGMEVL LINE EVALUATION ROUTINE RT-11 MMAC  VM02-10   14-OCT-76 01:37:52 PAGE 10+
SYMBOL TABLE

ABRFLG=  0040      AFAIL =  0030      ALLOK =  0004      ALLTG = XXXXXX G   AMAT  =  0010
ARRAY =  0020      ASGCOD= XXXXXX G   ASTR  =  0020      ATSNTG= XXXXXX G   A.STAT= XXXXXX G
AOX   = XXXXXX G   BAKSTG= XXXXXX G   BMAT  =  0004      BRKCNT= XXXXXX G   BRKCOD= XXXXXX G
BSTR  =  0008      CALLTG= XXXXXX G   CDOPTR= XXXXXX G   CDSPTR= XXXXXX G   CLASS    0145R
CLASSA   01D9R     CLASSB   01E4R     CLASSC   01F3R     CLASSD   01FFR     CLASSE   0207R
CLASST   015CR     CLASS1   015FR     CLASS2   0165R     CLASS3   015CR     CLASS4   0172R
CLASS5   0179R     CLASS6   0193R     CLASS7   0198R     CLASS8   01A4R     CLASS9   01B0R
CLPTR = XXXXXX G   CLSSA    0182R     CMAT  =  0001      CMDTBL= XXXXXX G   COMPR = XXXXXX G
CONCOD= XXXXXX G   CRCOD = XXXXXX G   CSTR  =  0002      CTCALL   01C4R     CTERR    0187R
CTKILL   01BFR     CTKN  = XXXXXX G   CTLR     01CDR     CTMM     01D6R     CTMS     01D3R
CTPULL   01C2R     CTSM     01DDR     CTSNGM   01CAR     DATCOD= XXXXXX G   DIMFLG=  0004
DISSRQ=  0008      DREXTA= XXXXXX G   DREXTB= XXXXXX G   DSPCHR= XXXXXX G   DSPLY = XXXXXX G
END   = XXXXXX G   EOFCOD= XXXXXX G   EOFTBL= XXXXXX G   EOLTG = XXXXXX G   EOSTG = XXXXXX G
EQUCOD= XXXXXX G   ERASGN= XXXXXX G   ERBRK = XXXXXX G   ERDOMN= XXXXXX G   EREOFN= XXXXXX G
ERFORA= XXXXXX G   ERLNNF= XXXXXX G   ERNDT = XXXXXX G   ERNIMX= XXXXXX G   ERNOFN= XXXXXX G
ERNXFN= XXXXXX G   ERNXTA= XXXXXX G   EROFA = XXXXXX G   ERRCD = XXXXXX G   ERRCDB= XXXXXX G
ERSHAR= XXXXXX G   ERSTOP= XXXXXX G   ERUNDF= XXXXXX G   ERVAL = XXXXXX G   ERWSFL= XXXXXX G
ESTG  = XXXXXX G   EXTFLG=  0080      FILES = XXXXXX G   FNACOD= XXXXXX G   FNFLG =  0010
FNTBL = XXXXXX G   FORTG = XXXXXX G   FUDGH = XXXXXX G   FUDGL = XXXXXX G   GLBFLG= XXXXXX G
GOSTG = XXXXXX G   GRAPH = XXXXXX G   IDLE  = XXXXXX G   IMACOD= XXXXXX G   IMXFLG=  0020
IMXTG = XXXXXX G   INAGTE= XXXXXX G   IOPROC= XXXXXX G   ITM1TG= XXXXXX G   ITM2TG= XXXXXX G
JMPAX = XXXXXX G   JMPX  = XXXXXX G   KEYFLG=  0010      KEYSTK= XXXXXX G   LBRKTG= XXXXXX G
LCLFLG= XXXXXX G   LDAX  = XXXXXX G   LDBX  = XXXXXX G   LDXX  = XXXXXX G   LISTTG= XXXXXX G
LITCOD= XXXXXX G   LITREL= XXXXXX G   LNCLOK   0084R     LNEND    005CR     LNEVL    0068RG
LNEXT    0016R     LNIMX    008AR     LNMCMP   0130R     LNNOTG= XXXXXX G   LNPDY    0091R
LNSIZE   0000R     LNSOK    0087R     LNSZOK   0013R     LNTRC    0220R     LNWFER   013FR
LNWSFL   0123R     LOCTG = XXXXXX G   LSP   = XXXXXX G   LSTCOD= XXXXXX G   MATMAT= XXXXXX G
MATSCL= XXXXXX G   MMICOD= XXXXXX G   MPLCOD= XXXXXX G   MSKCTR= XXXXXX G   MULCOD= XXXXXX G
NCSFLG=  0001      NLPTR = XXXXXX G   NTAPTR=  0008      NTATTR=  0004      NTDIMS=  0009
NTDLEN=  0005      NTLINK=  0000      NTNAME=  0002      NTPTR = XXXXXX G   NTSPTR=  0008
NTVAL =  0005      NTWCOL=  0007      NTWLEN=  0007      NTWROW=  0005      NULLTG= XXXXXX G
OBJATR=  0002      OBJBCK=  0003      OBJCOM   011DR     OBJDT =  0005      OBJFL    0142R
OBJGO    0087R     OBJGOB   00C0R     OBJLEN=  0000      OBJLOW   00F2R     OBJLP    008DR
OBJNFN   00E3R     OBJOPR   0104R     OCTBL = XXXXXX G   ONSFLG=  0002      ONTBL = XXXXXX G
OPRADR= XXXXXX G   OPRCL = XXXXXX G   OPRTBL= XXXXXX G   QUAFLG=  0002      PAETG = XXXXXX G
PARM  =  0008      PGMATR=  0002      PGMBP =  0005      PGMCO =  0009      PGMEVL   005FRG
PGMEXT   0057R     PGMFP =  0003      PGMHLT   0044R     PGMLEN=  0000      PGMLNN=  0007
PGMLP    006BR     PGMPTR= XXXXXX G   PGMRTN   005AR     PGMTG = XXXXXX G   PLOSTG= XXXXXX G
PNDEOF= XXXXXX G   PNDFLG= XXXXXX G   PNTNTG= XXXXXX G   PNTSTG= XXXXXX G   PRTTG = XXXXXX G
PSCTG = XXXXXX G   QTRNSL= XXXXXX G   QUSRFN= XXXXXX G   RFAIL =  00C0      RMAT  =  0040
RSTR  =  0080      RTRNTG= XXXXXX G   RUNFLG=  0080      R0    = XXXXXX G   R1    = XXXXXX G
R10   = XXXXXX G   R11   = XXXXXX G   R16   = XXXXXX G   R2    = XXXXXX G   R3    = XXXXXX G
R4    = XXXXXX G   R5    = XXXXXX G   R6    = XXXXXX G   R7    = XXXXXX G   R8    = XXXXXX G
R9    = XXXXXX G   SBP   = XXXXXX G   SCALER=  0040      SCLMAT= XXXXXX G   SEMITG= XXXXXX G
SETARG= XXXXXX G   SETUPB   0222R     SETUPD   0219R     SETUPF   0216R     SETUPI   0213R
SETUPL   021FR     SETUPP   0225R     SETUPT   021CR     SIZCOD= XXXXXX G   SIZERR= XXXXXX G
SLOPH = XXXXXX G   SLOPL = XXXXXX G   SNGMAT= XXXXXX G   STAX  = XXXXXX G   STPFLG=  0040
STPTR = XXXXXX G   STRING=  0010      SYSERR= XXXXXX G   TAPE  = XXXXXX G   TAPFIL= XXXXXX G
TRCEXT   0252R     TRCFLG=  0020      TRCLP    0241R     TSTINT= XXXXXX G   TYPARG= XXXXXX G
TYPRES= XXXXXX G   UNDEF =  0080      VALERR=  0040      VALTG = XXXXXX G   XEQSP = XXXXXX G
XEQSTR= XXXXXX G
. ABS.    0000      00
          0255      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2347. WORDS
.SY:PGMEVL/C(DK1:SEICLI,PGMEVL
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

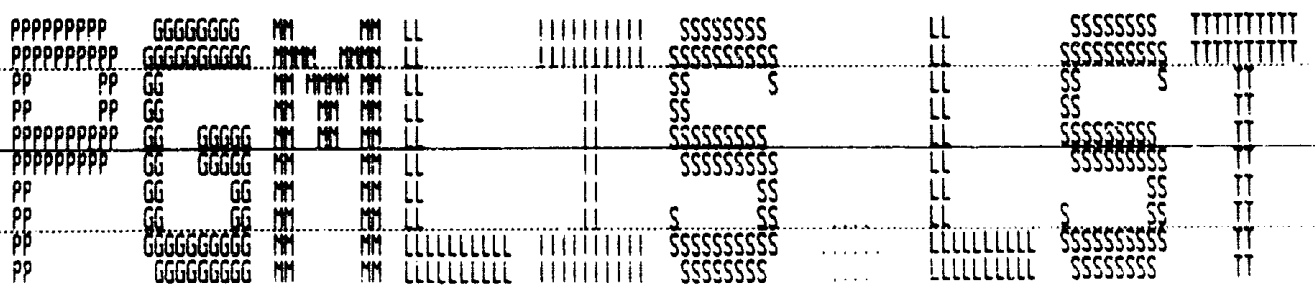
CROSS REFERENCE TABLE (CREF V01-03)

```
ONSFLG  2-14#   7-110
ONTBL   3-4#
OPRADR  2-24#   7-149*  7-159   7-168*  7-170   7-171   7-176*  9-31
OPRCL   7-6#    8-89
OPRTBL  7-8#    7-172
OUAFLG  2-22#
PBETG   5-15#
PARM    4-12#
PGMATR  4-37#
PGMBP   4-39#
PGMCD   4-41#   7-108
PGMEVL  7-4#    7-84#
PGMEXT  7-55    7-64    7-78#
PGMFP   4-38#   7-102   7-105
PGMHLT  7-52    7-57    7-61    7-69#
PGMLEN  4-36#
PGMLNN  4-40#   10-7
PGMLP   7-67    7-91#
PGMPTR  2-50#
PGMRTN  7-73    7-75    7-79#
PGMTG   5-10#
PLOSTG  5-6#
PNDEOF  2-47#
PNDFLG  2-33#   7-43    7-44*
PNTNTG  5-14#
PNTSTG  5-12#
PRTTG   5-26#
PSCTG   5-13#
QTRNSL  7-6#    7-72
QUSRFN  7-6#    7-74
R0      2-28#   7-48*   7-50    7-113*  7-118   7-119
R1      2-28#
R10     2-29#   7-12#   10-12*  10-14
R11     2-29#
R16     7-12#   10-10
R2      2-28#
R3      2-28#
R4      2-28#   8-11    8-19    8-49    8-139
R5      2-28#
R6      2-28#
R7      2-28#
R8      2-29#
R9      2-29#
RFAIL   5-35#
RMAT    5-34#   8-22    8-39    8-41    8-43    8-72    8-148
RSTR    5-33#   8-57    8-65    8-125
RTRNTG  5-25#
RUNFLG  2-17#   7-84
SBP     2-49#
SCALER  4-9#
SCLMAT  7-7#    8-102
SEMITG  5-21#
SETARG  7-5#    7-169
SETUPB  7-210   9-25#
SETUPD  7-210   9-13#
SETUPF  7-210   9-9#
SETUPI  7-210   9-5#
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

PLR0 1-20#
PLR8 1-27#
SEI 1-3#



```
                                .IDENT  /MCD505/
                                .GLOBL  PGMLIS
                        0000'   PGMLIS=.
                                .TITLE  PGMLIS T BUILDER
                                .GLOBL  R0,R1,R2,R3,R4,R5,ITM1TG,PGMPTR,KBFLAG
                                .GLOBL  GETSMA,DELET,PULFPN,PSHFPN,DISBLE,ENABLE


                        ;       THIS ROUTINE ADDS A LINE TO THE PROGRAM AT THE APPROPRIATE
                        ;       SPOT ACCORDING TO ITS LINE NUMBER, DELETING AN EXISTING
                        ;       LINE (THAT HAS THE SAME LINE NUMBER) IF NECESSARY.
                        ;
                        ;       INPUT:  X       -POINTER TO THE LINE (ALREADY IN USER MEMORY)
                        ;                        TO BE ADDED TO THE PROGRAM
                        ;
                        ;       OUTPUT:         -THE LINE INSERTED INTO THE PROGRAM.
                        ;
        0000    32              PUL A                   ; R1 IS THE RETURN ADDR
        0001    33              PUL B
        0002    97      00G     STA A   R1,D
        0004    D7      01G     STA B   (R1+1),D
        0006    DF      00G     STX     R2,D            ; R2 IS  TO THE LINE
        0008    EE      07      LDX     7,X
        000A    DF      00G     STX     R3,D            ; R3 IS THE LINE NUMBER
        000C    96      01G     LDA A   (R3+1),D
        000E    36              PSH A
        000F    96      00G     LDA A   R3,D
        0011    36              PSH A
        0012    B6      00G     LDA A   ITM1TG,I
        0014    36              PSH A
        0015    BD      0000G   JSR     GETSMA
        0018    BD      0000G   JSR     DISBLE
        0018    32              PUL A
        001C    32              PUL A                   ; R4 IS  TO PREDECESSOR
        001D    33              PUL B                   ; LINE
        001E    97      00G     STA A   R4,D
        0020    D7      01G     STA B   (R4+1),D
        0022    32              PUL A
        0023    32              PUL A                   ;R5 IS # OF PREDESSOR LINE
        0024    33              PUL B
        0025    97      00G     STA A   R5,D
        0027    D7      01G     STA B   (R5+1),D
        0029    DE      00G     LDX     R4,D
        0028    27      3A      BEQ     FIRST
        002D    DE      00G     LDX     R3,D
        002F    9C      00G     CPX     R5,D            ; DOES THIS LINE ALREADY EXIST
        0031    26      27      BNE     CHECK           ; IF NOT, DON'T NEED TO DELETE
                                                        ; ANYTHING; ELSE MUST DELETE
                                                        ; THE EXISTING COPY BEFORE
                                                        ; LINKING IN THE NEW ONE
        0033    DE      00G     LDX     R4,D            ; UPDATE THE PTR TO THE
        0035    EE      05      LDX     5,X             ; PREDECESSOR TO POINT TO THE
        0037    DF      00G     STX     R4,D            ; PREDECESSOR OF THE LINE WE'RE
                                                        ; GOING TO DELETE
        0039    CE      0000G   LDX     R0,I
        003C    BD      0000G   JSR     PSHFPN
```

```
0036    CE   0000G            LDX    R4,I
0042    BD   0000G            JSR    PSHFPN
0045    DE     00G            LDX    R5,D
0047    DF     00G            STX    R3,D
0049    DF     00G            STX    R4,D
004B    BD   0000G            JSR    DELET
004E    CE   0000G            LDX    R4,I
0051    BD   0000G            JSR    PULFPN
0054    CE   0000G            LDX    R0,I
0057    BD   0000G            JSR    PULFPN
005A    DE     00G   CHECK:   LDX    R2,D         ; NOW THAT WE'VE GOTTEN RID
005C    EE     00             LDX    0,X          ; OF THE OLD LINE, IS THE
005E    8C   000A             CPX    10.,I        ; NEW ONE WORTH LINKING
                                                  ; IN?   THAT IS, IS IT EMPTY?
                                                  ; IF SO RETURN
0061    27     5C             BEQ    DONE
0063    DE     00G   NEW:     LDX    R4,D         ; IF WE'RE INSERTING A NEW
0065    26     2A             BNE    INSERT       ; LINE DOES IT HAVE A PREDECESSOR
                                                  ; IF SO GO TO "INSERT"
0067    DE     00G   FIRST:   LDX    R2,D
0069    EE     00             LDX    0,X
006B    8C   000A             CPX    10.,I
006E    27     4F             BEQ    DONE
0070    DE     00G            LDX    R2,D         ; IF NOT PUT IT AT THE FRONT
0072    96     00G            LDA A  PGMPTR,D     ; OF THE PROGRAM
0074    A7     03             STA A  3,X          ; FORWARD PTR = PGM PTR
0076    96     01G            LDA A  (PGMPTR+1),D
0078    A7     04             STA A  4,X
007A    4F                    CLR A
007B    A7     05             STA A  5,X          ; BACK PTR = NULL
007D    A7     06             STA A  6,X
007F    96     00G            LDA A  R2,D
0081    97     00G            STA A  PGMPTR,D     ; PGM PTR =   TO LINE
0083    D6     01G            LDA B  (R2+1),D
0085    D7     01G            STA B  (PGMPTR+1),D
0087    EE     03             LDX    3,X          ;LOAD PTR TO SUCCESSOR
0089    27     04             BEQ    1$           ;IF NULL, QUIT
008B    A7     05             STA A  5,X          ;SET BACK PTR OF SUCCESSOR
008D    E7     06             STA B  6,X          ;LINE = TO   TO THE LINE
008F    20     2E    1$:      BRA    DONE
0091    DE     00G   INSERT:  LDX    R4,D         ; R5 IS NOW A TEMP PTR (PTR1)
0093    EE     03             LDX    3,X          ; R5 = FORWARD PTR OF PREDECESSOR
0095    DF     00G            STX    R5,D
0097    DE     00G            LDX    R4,D         ; FORWARD PTR OF PREDECESSOR:  TO LINE
0099    96     00G            LDA A  R2,D
009B    A7     03             STA A  3,X
009D    96     01G            LDA A  (R2+1),D
009F    A7     04             STA A  4,X
00A1    DE     00G            LDX    R5,D         ; BACK PTR OF SUCCESSOR (IF THERE IS
00A3    27     08             BEQ    1$
00A5    96     00G            LDA A  R2,D         ; ONE) =  TO LINE
00A7    A7     05             STA A  5,X
00A9    96     01G            LDA A  (R2+1),D
00AB    A7     06             STA A  6,X
00AD    DE     00G   1$:      LDX    R2,D         ; FORWARD PTR OF LINE =   TO SUCCESSOR
00AF    96     00G            LDA A  R5,D
00B1    A7     03             STA A  3,X
```

```
130  0083  96  01G          LDA A  (R5+1),D
131  0085  A7  04           STA A  4,X          ; BACK PTR OF LINE =  TO PREDECESSOR
132  0087  96  00G          LDA A  R4.D
133  0089  A7  05           STA A  5,X
134  008B  96  01G          LDA A  (R4+1),D
135  008D  A7  06           STA A  6,X
136  008F  BD  0000G  DONE: JSR    ENABLE
137  00C2  DE  00G          LDX    R1,D
138  00C4  6E  00           JMP    0,X
139       0001'             .END
```

```
PGMLIS T BUILDER       RT-11 MMAC  VM02-10   14-OCT-76 01:38:11 PAGE 1+
SYMBOL TABLE

CHECK     005AR       DELET = xxxxxx G      DISBLE= xxxxxx G      DONE      008FR       ENABLE= xxxxxx G
FIRST     0067R       GETSMA= xxxxxx G      INSERT    0091R       ITMITG= xxxxxx G      KBFLAG= xxxxxx G
NEW       0063R       PGMLIS=   0000RG      PGMPTR= xxxxxx G      PSHFPN= xxxxxx G      PULFPN= xxxxxx G
R0    = xxxxxx G      R1    = xxxxxx G      R2    = xxxxxx G      R3    = xxxxxx G      R4    = xxxxxx G
R5    = xxxxxx G
. ABS.    0000        00
          00C6        01
ERRORS DETECTED: 0    WARNINGS POSTED: 0    FREE CORE: 3389. WORDS
.SY:PGMLIS/C(DK1:SETCLI,PGMLIS
```

CROSS REFERENCE TABLE (CREF V01-03.)

1-18

CHECK 1-63 1-83#
DELET 1-21# 1-78
DISBLE 1-21# 1-48
DONE 1-88 1-95 1-112 1-136#
ENABLE 1-21# 1-136
FIRST 1-60 1-92#
GETSWA 1-21# 1-47
INSERT 1-90 1-113#
LTM1TG 1-20# 1-45
KBFLAG 1-20#
NEW 1-89#
PGMLIS 1-17# 1-18#
PGMPTR 1-20# 1-97 1-99 1-105* 1-107*
PSHFPN 1-21# 1-72 1-74
PULFPN 1-21# 1-80 1-82
R0 1-20# 1-71 1-81
R1 1-20# 1-36* 1-37* 1-137
R2 1-20# 1-38* 1-83 1-92 1-96 1-104 1-106 1-117 1-119 1-123 1-125 1-127
R3 1-20# 1-40* 1-41 1-43 1-61 1-76*
R4 1-20# 1-52* 1-53* 1-59 1-67 1-69* 1-73 1-77* 1-79 1-89 1-113 1-116 1-132 1-134
R5 1-20# 1-57* 1-58* 1-62 1-75 1-115* 1-121 1-128 1-130

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3#

```
PPPPPPPPP   RRRRRRRRR   IIIIIIIIII  NN      NN  TTTTTTTTTT  FFFFFFFFFF          LL          SSSSSSSS   TTTTTTTTTT
PPPPPPPPPP  RRRRRRRRRR  IIIIIIIIII  NNN     NN  TTTTTTTTTT  FFFFFFFFFF          LL          SSSSSSSSSS  TTTTTTTTTT
PP      PP  RR      RR      II      NN N    NN      TT      FF                  LL          SS      S       TT
PP      PP  RR      RR      II      NN NN   NN      TT      FF                  LL          SS              TT
PPPPPPPPPP  RRRRRRRRRR      II      NN  NN  NN      TT      FFFFFF              LL          SSSSSSSS        TT
PPPPPPPPP   RRRRRRRRR       II      NN  NN  NN      TT      FFFFFF              LL           SSSSSSSS       TT
PP          RR   RR         II      NN   NN NN      TT      FF                  LL                SS       TT
PP          RR    RR        II      NN    N NN      TT      FF                  LL          S     SS       TT
PP          RR     RR  IIIIIIIIII  NN     NNN      TT      FF          ......  LLLLLLLLLL  SSSSSSSSS      TT
PP          RR      RR  IIIIIIIIII  NN      NN      TT      FF          ....    LLLLLLLLLL   SSSSSSS       TT
```

```
                                .TITLE  PRINTF    PRINT STATEMENT FORMATTING.
                                .GLOBL  PRINTF
0000'                           PRINTF=.
                                .IDENT  /JKP008/

                                .GLOBL  ERNFST          ; NO FORMAT STRING.
                                .GLOBL  ERINFS          ; OUT OF FORMAT STRING.
                                .GLOBL  ERIFC           ; ILLEGAL FORMAT CHAR.
                                .GLOBL  ERINU           ; ILLEGAL NUMBER USAGE.
                                .GLOBL  ERTMDS          ; TOO MANY DATA SPECIFIERS
                                .GLOBL  ERBPNU          ; BAD PARENTHESES USAGE.
                                .GLOBL  ERBMDU          ; BAD MODIFIER USAGE.
                                .GLOBL  ERLRSU          ; BAD LRS USAGE
                                .GLOBL  ERBCMU          ; BAD COMMA USAGE
                                .GLOBL  ERBDPU          ; BAD DECIMAL POINT USAGE
                                .GLOBL  ERQUOT          ; INCOMPLETE LITERAL
                                .GLOBL  ERDATM          ; DATA TYPE MISMATCH
                                .GLOBL  ERFLDO          ; FIELD OVERFLOW
                                .GLOBL  ERDEOR          ; EXPONENT OUT OF RANGE FOR 'D' OPERATOR
                                .GLOBL  ERTABO          ; TABBING ERROR.

                                .GLOBL  RTRNTG          ; RETURN FLAG.

                                .GLOBL  FMTINI          ; INITIALIZE ROUTINE.
                                .GLOBL  FMTPNT          ; DATA ITEM OUTPUT
                                .GLOBL  FMTCLN          ; FORMAT STRING CLEAN-UP.
                                .GLOBL  PUTBYT          ; PUT BYTE TO OUTPUT DEV.
                                .GLOBL  RNDATA          ; DECIMAL ROUND AT 'A'.
                                .GLOBL  DEFPNT          ; DEFAULT PRINT.
                                .GLOBL  UPCASE          ; MAKE LOWER CASE UPPER CASE.

                                .GLOBL  ERRCD           ; ERROR BYTE.

                                .GLOBL  L1              ; REPEAT LOOP1
                                .GLOBL  SP1             ; RELATIVE STRING POINTER 1.
                                .GLOBL  L2              ; REPEAT LOOP 2.
                                .GLOBL  SP2             ; RELATIVE STRING POINTER 2.
                                .GLOBL  L3              ; REPEAT LOOP 3.
                                .GLOBL  SP3             ; RELATIVE STRING POINTER 3.
                                .GLOBL  L4              ; REPEAT LOOP 4.
                                .GLOBL  SP4             ; RELATIVE STRING POINTER 4.

                                .GLOBL  CURSER          ; RELATIVE CURSER POINTER.
                                .GLOBL  PARCNT          ; PARENTHESIS COUNT.
                                .GLOBL  TABCNT          ; CURRENT LOGICAL OUTPUT POSITION.

                                .GLOBL  ISP             ; IMAGE STRING POINTER.
                                .GLOBL  ISL             ; IMAGE STRING LENGTH.
                                .GLOBL  DP              ; DATA POINTER.
                                .GLOBL  DL              ; DATA LENGTH.
                                .GLOBL  DT              ; DATA TYPE.

                                .GLOBL  LITFLG          ; LITERAL FLAG.
                                .GLOBL  LRSFLG          ; LRS SUPPRESS FLAG.
                                .GLOBL  NUMFLG          ; NUMBER FLAG.
                                .GLOBL  FFLG            ; AS REQUIRED FLAG.
```

```
                        .GLOBL  PFLG            ; PLUS FLAG.
                        .GLOBL  MFLG            ; MINUS FLAG.
                        .GLOBL  DOLF            ; DOLLAR FLAG.
                        .GLOBL  COMF            ; COMMA FORMAT FLAG.
                        .GLOBL  FMTFLG          ; FORMAT TYPE IN PROCESS.
                                                ; 0=FMTOUT.
                                                ; NOT 0=FMTCLN.
                        .GLOBL  FLAGS           ; 8 FLAGS (SPNE+-$C).
```

```
                        .GLOBL  K0              ; PAGE 0 WORKING AREA
                        .GLOBL  K1
                        .GLOBL  K2
                        .GLOBL  K3
                        .GLOBL  K4
                        .GLOBL  K5
                        .GLOBL  K6


                        .GLOBL  ES              ; EXPONENT SIGN
                        .GLOBL  E1              ; EXPONENT LSD
                        .GLOBL  E2
                        .GLOBL  E3              ; EXPONENT MSD
                        .GLOBL  N1              ; NUMBER LSD.
                        .GLOBL  N11
                        .GLOBL  N12             ; NUMBER MSD
                        .GLOBL  NS              ; NUMBER SIGN
```

```
                        .GLOBL  X0              ; FP FRACTION.
                        .GLOBL  X1
                        .GLOBL  X2
                        .GLOBL  X3
                        .GLOBL  X4
                        .GLOBL  X5

                        .GLOBL  DDP             ; DIGITS DECIMAL POINT.
                        .GLOBL  DN1             ; NUMBER OF LEFT DIGIT.
                        .GLOBL  DN2             ; NUMBER OF RIGHT DIGITS.
                        .GLOBL  DISCNT          ; NUMBER OF OUTPUT DIGITS.
                        .GLOBL  F1              ; AS REQUIRED LEFT.
                        .GLOBL  F2              ; AS REQUIRED RIGHT.
                        .GLOBL  EXH             ; EXPONENT (HIGH ORDER).
                        .GLOBL  EXL             ; EXPONENT (LOW ORDER).
                        .GLOBL  DPCONT          ; DECIMAL POINT CONTROL.
                        .GLOBL  EXP             ; CONDENSED EXPONENT.
                        .GLOBL  EXS             ; EXPONENT BINARY SIGN.

                        .GLOBL  LSPS            ; LEADING SPACES.
                        .GLOBL  LDIG            ; LEFT DIGIT COUNT.
                        .GLOBL  LDIGA           ; LEFT DIGIT BEGINNING ADDRESS.
                        .GLOBL  LCOM            ; COMMA COUNT.
                        .GLOBL  ZERO1           ; LEFT TRAILING ZEROS.
                        .GLOBL  ZERO2           ; RIGHT LEADING ZERO COUNT.
                        .GLOBL  RDIG            ; RIGHT PRINTING DIGITS COUNT.
                        .GLOBL  RDIGA           ; RIGHT PRINT POINTER.
                        .GLOBL  ZERO3           ; RIGHT TRAILING ZERO COUNT.

                        .GLOBL  ZERCNT          ; COMMON ZERO COUNT.
                        .GLOBL  ZERSVH          ; ZERO SAVE ADDRESS HIGH.
                        .GLOBL  ZERSVL          ; ZERO SAVE ADDRESS LOW.

                        .GLOBL  COMCNT          ;COMMA REMAINDER COUNT.
```

```
                                
                                
0000  7F  0000G  FMTPNT: CLR     DPCONT          ; CLEAR DECIMAL CONTROL.
0003  86  000G           LDA A   RTRNTG,I        ; TAG RETURN.
0005  36                 PSH A
0006  BD  007F'          JSR     FMTOUT          ; OUTPUT DATA TYPE.
0009  32                 PUL A                   ; SCRATCH RETURN.
000A  39                 RTS                     ; RETURN.
                                
                                
000B  86  FF     DEFPNT: LDA A   377,I           ; CONTROL=DEFAULT CK.
000D  B7  0000G          STA A   DPCONT
0010  86  000G           LDA A   RTRNTG,I        ; TAG RETURN.
0012  36                 PSH A
0013  96  000G           LDA A   DT,D            ; GET DATA TYPE.
0015  81  01             CMP A   1,I
0017  27  12             BEQ     DFNUM           ; IT IS NUMBER.
0019  CE  0056'          LDX     APNT,I          ; GET IMAGE STRING ADDRESS.
001C  DF  000G           STX     ISP,D           ; SET UP FMTOUT.
001E  CE  0002           LDX     2,I             ; GET IMAGE COUNT.
0021  DF  000G   DEFFIN: STX     ISL,D           ; SET UP FMTOUT.
0023  BD  005F'          JSR     FMTINI          ; INITIALIZE FMTOUT.
0026  BD  007F'          JSR     FMTOUT          ; OUTPUT DATA.
0029  32                 PUL A                   ; SCRATCH TAG.
002A  39                 RTS                     ; RETURN
002B  B6  0000G  DFNUM:  LDA A   N12             ; GET D INDICATOR
002E  27  0C             BEQ     NRMOUT          ; 0, OUTPUT NORMAL.
0030  D6  000G           LDA B   EXL,D           ; GET EXPONENT.
0032  96  000G           LDA A   EXH,D
0034  2B  10             BMI     DFM             ; NEGATIVE, CK DECIMAL.
0036  C0  07             SUB B   7.,I            ; EXP-7.
0038  82  00             SBC A   0,I
003A  24  10             BCC     SFOUT           ; + RESULT, DO SF OUTPUT.
003C  CE  005A'  NRMOUT: LDX     NPNT,I          ; GET NORMAL IMAGE ADDRESS.
003F  DF  000G           STX     ISP,D           ; SET UP FMTOUT.
0041  CE  0005           LDX     5,I             ; GET IMAGE LENGTH.
0044  20  DB             BRA     DEFFIN          ; FINISH.
0046  CB  03     DFM:    ADD B   3,I             ; 3+(-EXP).
0048  89  00             ADC A   0,I
004A  25  F0             BCS     NRMOUT          ; CARRY MEANS NORMAL OUTPUT.
004C  CE  0058'  SFOUT:  LDX     SFPNT,I         ; GET SF IMAGE STRING.
004F  DF  000G           STX     ISP,D           ; SET UP FMTOUT.
0051  CE  0002           LDX     2,I             ; GET IMAGE LENGTH.
0054  20  CB             BRA     DEFFIN          ; FINISH.
                                
                                
0056  46         APNT:   .BYTE   'F              ; CHARACTER IMAGE STRING.
0057  41                 .BYTE   'A
                                
                                
0058  46         SFPNT:  .BYTE   'F              ; SCIENTIFIC IMAGE STRING.
0059  45                 .BYTE   'E
                                
                                
005A  46         NPNT:   .BYTE   'F              ; NORMAL IMAGE STRING.
005B  44                 .BYTE   'D
005C  2E                 .BYTE   '.
```

```
    58        005D    46                      BYTE    'F
    59        005E    44                      BYTE    'D
```

```
                                        ; FMTINI IS THE INITIALIZE.  IT MUST BE CALLED
                                        ; BEFORE FMTOUT OR FMTCLN.  FMTINI CLEARS ALL
                                        ; FLAGS, SETS CURSER = TO FIRST CHARACTER, CHECKS FOR
                                        ; NULL STRING, THEN RETURNS.
                                        
                                        
005F  4F          FMTINI: CLR A                   ; A=0.
0060  97  00G             STA A  PARCNT,D         ; CLEAR ALL FLAGS.
0062  97  00G             STA A  LITFLG,D
0064  97  00G             STA A  LRSFLG,D
0066  97  00G             STA A  NUMFLG,D
0068  97  00G             STA A  FFLG,D
006A  97  00G             STA A  PFLG,D
006C  97  00G             STA A  MFLG,D
006E  97  00G             STA A  DOLFLG,D
0070  97  00G             STA A  COMFLG,D
0072  97  00G             STA A  CURSER,D         ; CURSER = FIRST CHARACTER.
0074  97  01G             STA A  CURSER+1,D
0076  DE  00G             LDX    ISL,D            ; CHECK FOR NULL STRING.
0078  26  04              BNE    BQ
007A  86  00G             LDA A  ERRFST,I         ; NULL STRING, SO INDICATE.
007C  97  00G             STA A  ERRCD,D
007E  39          BQ:     RTS                     ; RETURN.
```

```
                                        ; FMTOUT IS CALLED ONLY WHEN THERE IS A VALID DATA
                                        ; TYPE TO BE FORMATTED. FMTOUT IS CALLED REPETITIVELY
                                        ; UNTIL ALL DATA TYPES ARE EXHAUSTED. FMTOUT WILL
                                        ; AUTOMATICALLY OUTPUT NON-DATA TYPE SPECIFIERS WHILE
                                        ; SEARCHING FOR A MATCHING DATA TYPE SPECIFIER.
                                        ; BEFORE A SERIES OF CALLS ARE MODE TO FMTOUT, FMTINI
                                        ; MUST BE CALLED FIRST TO INITIALIZE THE FORMAT STRING
                                        ; ANALYZE PROCESS.

007E  86  000   FMTOUT: LDA A  #TRNTG,1  ; TAG RETURN.
0081  36                PSH A
0082  5F                CLR B           ; SET FMTOUT FLAG.
0083  D7  000           STA B  FMTFLG,D
0085  8D  31A6'  RECUR1: JSR   GETCHI    ; GET CHARACTER.
0088  25  35             BCS   SEARCH    ; GOT SOMETHING, CONTINUE.
008A  86  000           LDA A  #ERLNS,1  ; GOT NOTHING, GE. ERROR BYTE.
008C  97  000   EREXIT: STA A  ERRCD,D   ; SET ERROR, STRING TOO SHORT.
008E  32        EXIT:   PUL A           ; SCRATCH TAG.
008F  39                RTS             ; RETURN.
```

```
                                        ; FMTCLN IS USED TO OUTPUT ANY REMAINING
                                        ; SPECIFIERS THAT DO NOT REQUIRE DATA ITEMS FROM
                                        ; THE PRINT LIST.  IT IS CALLED EITHER FOR A NULL PRINT
                                        ; LIST OR AFTER THE PRINT LIST IS EXHAUSTED TO CLEAN-UP
                                        ; AND CLOSE OUT THE FORMAT STRING.  BEFORE ANY CALLS TO
                                        ; FMTCLN ARE MADE, FMTINI MUST BE CALLED TO
                                        ; INITIALIZE THE PROCESS.

0090  96  000   FMTCLN: LDA A  ERRCD.D   ; GET ERROR CODE.
0092  27  01              BEQ    FN       ; 0=OK TO CONTINUE.
0094  39                  RTS             ; ERROR, GET OUT IMMEDIATELY.
0095  86  000   FN:       LDA A  RTRNTG.I ; TAG RETURN.
0097  36                  PSH A
0098  86  FF              LDA A  377.I    ; SET FMTCLN FLAG.
009A  97  000             STA A  FMTFLG.D
009C  BD  0'A6'  RECUR2:  JSR    GETCHI   ; GET CHARACTER.
009F  25  1E              BCS    SEARCH   ; GOT SOMETHING, CONTINUE.
00A1  BD  01C3'           JSR    FLAGS    ; CONDENCE FLAGS.
00A4  C5  7F              BIT B  177.I    ; MASK OFF LRS.
00A6  27  0C              BEQ    B8       ; NO MODIFIERS,  CK LRS.
00A8  C5  3F              BIT B  77.I     ; EXTRACT MODIFIER FLAGS.
00AA  27  04              BEQ    B13      ; NOT SET, MUST BE PAREN.
00AC  86  000             LDA A  ERBMDU.I ; GET MODIFIER ERROR FLAG.
00AE  20  0C    CLNER:    BRA    EREXIT   ; FLAG ERRORS.
0080  86  000   B13:      LDA A  ERBPNU.I ; GET PAREN ERROR FLAG.
00B2  20  FA              BRA    CLNER    ; FINISH.
00B4  96  000   B8:       LDA A  LRSFLG.D ; GET SUPPRESS FLAG.
00B6  25  D6              BNE    EXIT     ; SUPPRESS OUTPUT.
00B8  86  0D    B9:       LDA A  15.I     ; GET CR.
00BA  BD  01B4'           JSR    PUTCHR   ; OUTPUT IT.
00BD  20  CF              BRA    EXIT     ; RETURN.
```

```
                                        ; SEARCH IS USED TO SEARCH THE TABLE FOR A VALID
                                        ; FORMAT STRING FIELD OPERATOR OR MODIFIER CHARACTER.
                                        ; THE TABLE SEARCH IS VALID FOR ALL CHARACTERS EXCEPT
                                        ; NUMBERS WHICH ARE HANDLED SEPARATELY.  IF
                                        ; SEARCH DOES NOT FIND A VALID CHARACTER, IT FLAGS
                                        ; THE ERROR AND RETURNS TO CALLER.  IF A VALID
                                        ; CHARACTER IS FOUND, CONTROL PASSES TO THE
                                        ; APPROPRIATE CHARACTER HANDLER ROUTINE.  ON
                                        ; ENTRY TO SEARCH, THE CHARACTER IN QUESTION
                                        ; MUST BE IN 'A'.  SEARCH IS NOT A SUBROUTINE
                                        ; BUT A "FALL THROUGH" ROUTINE WHICH FMTOUT
                                        ; OR FMTCLN BRANCH TO.

00BE  CE  00E7'   SEARCH: LDX   TABLE,I   ; GET TABLE POINTER.
00C2  C6  14              LDA B 20.,I     ; B=LOOP COUNT (20 ENTRIES).
00C4  A1  00      SLOOP:  CMP A 0,X       ; COMPARE?
00C6  26  04              BNE   B2        ; NO.  TRY NEXT ITEM.
00C8  EE  01              LDX   1,X       ; GET HANDLER ADDRESS.
00CA  6E  00              JMP   0,X       ; DO FORMAT SPECIFIER.
00CC  5A          B2:     DEC B           ; BUMP DOWN COUNT.
00CD  27  05              BEQ   B3        ; ALL ITEMS EXHAUSTED, TRY NUMBER.
00CF  08                  INX             ; BUMP POINTER TO NEXT ITEM.
00D0  08                  INX
00D1  08                  INX
00D2  20  F0              BRA   SLOOP     ; CONTINUE LOOP.
00D4  BD  0177'   B3:     JSR   CKNUM     ; CHAR=0-9?
00D7  25  04              BCS   GETN      ; GET A NUMBER.
00D9  86  00G             LDA A ERIFC,I   ; ILLEGAL FORMAT CHAR.
00DB  20  AF              BRA   EREXIT    ; TAKE ERROR EXIT.
00DD  BD  0123'   GETN:   JSR   GETNUM    ; GET A NUMBER.
00E0  B6  0000G   RECURS: LDA A FMTFLG    ; GET FMT MODE FLAG.
00E3  26  B7              BNE   RECUR2    ; RECURSE TO FMTCLN.
00E5  20  9E              BRA   RECUR1    ; RECURSE TO FMTOUT.

                                        ; EACH ITEM IN THE TABLE IS OF THE FORM:
                                        ; FORMAT CHARACTER:  APPROPRIATE HANDLER ADDRESS.
                                        ; EACH ENTRY REQUIRES 3 BYTES.
                                        ; THE TABLE HANDLES ALL FORMAT STRING CHARACTERS
                                        ; EXCEPT NUMBERS.


00E7  41          TABLE:  .BYTE   'A
00E8  034E'               .WORD   CS      ; CHARACTER STRING
00EA  44                  .BYTE   'D
00EB  04E3'               .WORD   DI      ; DIGIT
00ED  45                  .BYTE   'E
00EE  0304'               .WORD   SF      ; SCIENTIFIC FORMAT
00F0  22                  .BYTE   '"
00F1  0311'               .WORD   LI      ; LITERAL
00F3  58                  .BYTE   'X
00F4  0240'               .WORD   SP      ; SPACES
00F6  2F                  .BYTE   '/
00F7  0268'               .WORD   CR      ; CARRIAGE RETURNS
00F9  4C                  .BYTE   'L
00FA  0264'               .WORD   LF      ; LINE FEEDS
00FC  53                  .BYTE   'S
00FD  0288'               .WORD   SCR     ; SUPPRESS CR/LF
```

```
                00FF    28              .BYTE   '(
                0100    0297'           .WORD   RS              ; REPEAT SPECIFIER
                0102    29              .BYTE   ')
                0103    02D7'           .WORD   ERF             ; END REPEAT FIELD
                0105    50              .BYTE   'P
                0106    0260'           .WORD   NP              ; NEW PAGE
                0108    2C              .BYTE   ',
                0109    0242'           .WORD   NOP             ; NO-OP
                010B    46              .BYTE   'F
                010C    0200'           .WORD   ASR             ; AS REQUIRED
                010E    2B              .BYTE   '+
                010F    020C'           .WORD   PLUS            ; PLUS.
                0111    2D              .BYTE   '-
                0112    0217'           .WORD   MINUS           ; MINUS.
                0114    2E              .BYTE   '.
                0115    0222'           .WORD   PERIOD          ; PERIOD.
                0117    24              .BYTE   '$
                0118    0227'           .WORD   DOLLAR          ; DOLLAR.
                011A    43              .BYTE   'C
                011B    0233'           .WORD   COMMA           ; COMMA MODIFIER.
                011D    20              .BYTE   40              ; (SPACE).
                011E    00E0'           .WORD   RECURS          ; IGNORE IMBEDDED SPACES.
                0120    54              .BYTE   'T
                0121    026C'           .WORD   TAB             ; TAB TO POSITION.
```

```
                                        ; GETNUM IS THE ROUTINE THAT GETS A NUMBER,
                                        ; OR TRIES TO.  IT ALLOWS UP TO THREE DECIMAL
                                        ; CHARACTERS AND ERROR EXITS IF THE NUMBER IS LESS
                                        ; THAN 1 OR GREATER THAN 255; IT ALSO ERRORS
                                        ; IF THE NUMBER IS THE LAST ITEM IN THE FORMAT STRING OR IF
                                        ; THERE HAS BEEN A PREVIOUS NUMBER ENTERED AND NOT USED.
                                        ; ON ENTRY, THE FIRST NUMBER, IN BCD, MUST BE
                                        ; IN 'A'.  FOR A NORMAL EXIT, THE NUMBER IS
                                        ; LEFT IN NUMFLG.

0123  D6  00G     GETNUM: LDA B   NUMFLG,D        ; GET NUMBER FLAG
0125  26  30              BNE     NUMERR          ; ALREADY SET, ERROR.
0127  D6  00G             LDA B   FFLG,D          ; GET F FLAG
0129  26  28              BNE     MDERR           ; ALREADY SET CONFLICT
012B  97  00G             STA A   NUMFLG,D        ; SAVE FIRST #.
012D  8D  56              BSR     GETCH           ; LOOK AHEAD.
012F  24  26              BCC     NUMERR          ; OUT OF CHARACTERS AT WRONG TIME.
0131  8D  44              BSR     CKNUM           ; IS IT NUMBER?
0133  24  19              BCC     NEXIT           ; NO. TRY EXIT
0135  8D  27              BSR     MUL10           ; INCLUDE IT.
0137  8D  01AE'           JSR     INCCUR          ; TAKE LAST CHARACTER.
013A  8D  49              BSR     GETCH           ; LOOK AHEAD.
013C  24  19              BCC     NUMERR          ; OUT OF CHARACTERS AT WRONG TIME.
013E  8D  37              BSR     CKNUM           ; IS IT NUMBER.
0140  24  0C              BCC     NEXIT           ; NO. TRY EXIT
0142  8D  1A              BSR     MUL10           ; INCLUDE IT.
0144  8D  68              BSR     INCCUR          ; TAKE LAST CHARACTER.
0146  8D  3D              BSR     GETCH           ; LOOK AHEAD.
0148  24  00              BCC     NUMERR          ; OUT OF CHARACTERS AT WRONG TIME.
014A  8D  2B              BSR     CKNUM           ; IS IT NUMBER?
014C  25  09              BCS     NUMERR          ; YES, TOO MANY DIGITS
014E  96  00G     NEXIT:  LDA A   NUMFLG,D        ; GET NUMBER
0150  27  05              BEQ     NUMERR          ; 0=NUMBER ERROR
0152  39                 RTS                     ; SUCCESS RETURN
0153  86  00G     MDERR:  LDA A   ERBMDU,I        ; CANNOT HAVE F&N AT SAME TIME
0155  20  02              BRA     MDNER           ; SET ERROR
0157  86  00G     NUMERR: LDA A   ERINU,I         ; GET NUMBER ERROR FLAG
0159  33          MDNER:  PUL B                   ; SCRATCH RETURN
015A  33                  PUL B
015B  7E  008C'           JMP     EREXIT          ; ERROR RETURN

015E  D6  00G     MUL10:  LDA B   NUMFLG,D        ; GET PREVIOUS NUMBER
0160  58                  ASL A                   ; MULTIPLY BY 2
0161  25  10              BCS     MULER           ; OVERFLOW ERROR
0163  58                  ASL B                   ; MULTIPLY BY 4
0164  25  0D              BCS     MULER           ; OVERFLOW ERROR
0166  DB  00G             ADD B   NUMFLG,D        ; MULTIPLY BY 5
0168  25  09              BCS     MULER           ; OVERFLOW ERROR
016A  58                  ASL B                   ; MULTIPLY BY 10
016B  25  06              BCS     MULER           ; OVERFLOW ERROR
016D  1B                  ABA                     ; ADD IN NEW DIGIT
016E  25  03              BCS     MULER           ; OVERFLOW ERROR
0170  97  00G             STA A   NUMFLG,D        ; SAVE RESULT
0172  39                  RTS                     ; RETURN
```

```
                0173  33              MULER:  PUL B                   ; SCRATCH RETURN
                0174  33                      PUL B
                0175  20  E0                  BRA     NUMERR          ; ERROR EXIT

                                              ; CKNUM CHECKS TO SEE IF THE ASCII CHARACTER IN
                                              ; 'A' IS WITHIN THE RANGE OF 60 - 71 OCTAL.  IF IT
                                              ; IS, THE CHARACTER IN 'A' IS REDUCED TO A
                                              ; BCD CHARACTER AND CARRY IS SET; IF IT IS NOT,
                                              ; 'A' IS UNCHANGED AND CARRY IS 0 ON EXIT.

                0177  81  30        CKNUM:  CMP A   60,I            ; A-60 LOWER LIMIT.
                0179  24  02                  BCC     B5              ; LOWER LIMIT OK.  CK HIGH
                017B  0C                      CLC                     ; NOT NUMBER, C=0.
                017C  39              B4:     RTS                     ; NOT NUMBER RETURN.
                017D  81  38          B5:     CMP A   73.I            ; OVER 72?
                017F  24  FB                  BCC     B4              ; YES, NOT NUMBER RETURN.
                0181  80  30                  SUB A   60,I            ; MAKE BCD.
                0183  00                      SEC                     ; SET C FOR NUMBER RETURN.
                0184  39                      RTS                     ; RETURN.
```

```
                                        ; GETCH GETS A CHARACTER FROM THE FORMAT STRING
                                        ; IN 'A' AND SETS CARRY = 1; IF THERE ARE NO
                                        ; MORE CHARACTERS, ON EXIT, C = 0 AND 'A' IS
                                        ; UNDEFINED.  GETCH DOES NOT INCREMENT THE CURSER.


                                        ; GETCHI CALLS GETCH AND THEN CALLS INCCUR
                                        ; TO INCREMENT (ADVANCE) THE CURSER TO THE
                                        ; NEXT CHARACTER POSITION.

                                        ; INCCUR INCREMENTS THE CURSER BY 1 SO
                                        ; IT IS ADVANCED TO  POINT TO THE NEXT
                                        ; CHARACTER IN LINE.

0185  DE  00G     GETCH:  LDX   CURSER,D        ; COMPARE CURSER POSITION
0187  9C  00G             CPX   ISL,D           ; TO IMAGE STATEMENT LENGTH.
0189  26  02              BNE   B1              ; NOT END, GET CHAR.
018B  0C                  CLC
018C  39                  RTS                   ; NO MORE CHAR'S RETURN.
018D  96  00G     B1:     LDA A CURSER,D        ; MAKE CURSER ABSOLUTE
018F  D6  01G             LDA B CURSER+1,D
0191  D8  01G             ADD B ISP+1,D
0193  99  00G             ADC A ISP,D
0195  97  00G             STA A K0,D            ; SAVE ABSOLUTE ADDRESS.
0197  D7  00G             STA B K1,D
0199  DE  00G             LDX   K0,D            ; SET INDEX REG AS POINTER.
019B  A6  00              LDA A 0,X             ; GET CHARACTER IN A.
019D  D6  00G             LDA B LITFLG,D        ; GET LITERAL FLAG.
019F  26  03              BNE   B111            ; IN LITERAL, SKIP UPCASE.
01A1  8D  0000G           JSR   UPCASE          ; UPCASE CHARACTER.
01A4  0D          B111:   SEC                   ; C = 1.
01A5  39                  RTS                   ; RETURN.

01A6  8D  DD      GETCH1: BSR   GETCH           ; LOOK FOR CHARACTER.
01A8  25  01              BCS   B6              ; FINISH.
01AA  39                  RTS                   ; NONE LEFT.
01AB  8D  01      B6:     BSR   INCCUR          ; INCREMENT CURSER
01AD  39                  RTS                   ; RETURN

01AE  DE  00G     INCCUR: LDX   CURSER,D        ; GER CURSER.
01B0  08                  INX                   ; INCREMENT IT.
01B1  DF  00G             STX   CURSER,D        ; RESTORE NEW VALUE.
01B3  39                  RTS
```

```
                                        ; PUTCHR OUTPUTS THE CHARACTER IN 'A'
                                        ; USING PUTBYT.  IF PUTBYT RETURNS WITH
                                        ; AN ERROR, AN ERROR EXIT IS MADE; OTHERWISE
                                        ; A NORMAL RETURN IS MADE TO THE INTERNAL CALLER.
                                        
0184    C6    000     PUTCHR: LDA B   RTRNTG,I        ; GET RETURN TAG.
0186    37                    PSH B                     ; TAG RETURN.
0187    BD    00000           JSR     PUTBYT          ; OUTPUT ASCII.
018A    96    000             LDA A   ERRCD,D         ; ANY ERRORS?
018C    27    03              BEQ     PTR             ; NO. NORMAL RETURN.
018E    32                    PUL A                   ; YES, POP PUTCHR RETURN.
018F    32                    PUL A
01C0    32                    PUL A
01C1    32            PTR:    PUL A                   ; POP TAG
01C2    39                    RTS                     ; RETURN
```

```
                                        ; FLAG8 CONDENSES THE FLAGS SO THEY ARE
                                        ; IN FLAGS IN THE FOLLOWING ORDER (0-7)
                                        ; LRSFLG,PARCNT,NUMFLG,FFLG,PFLG,MFLG,DOLFLG,COMFLG
                                        ; ON EXIT, FLAGS IS ALSO IN 'B'.

01C3  5F            FLAG8:  CLR B               ; CLEAR FLAG BYTE.
01C4  96  000       LDA A   LRSFLG,D            ; GET SUPPRESS FLAG.
01C6  80  1F        BSR     FLG1                ; CONDENSE IT.
01C8  96  000       LDA A   PARCNT,D            ; GET PARENTHESIS COUNT.
01CA  80  18        BSR     FLG1                ; CONDENSE IT.
01CC  96  000       LDA A   NUMFLG,D            ; GET NUMBER.
01CE  80  17        BSR     FLG1                ; CONDENSE IT.
01D0  96  000       LDA A   FFLG,D              ; GET "RS REQUIRED" FLAG.
01D2  80  13        BSR     FLG1                ; CONDENSE IT.
01D4  96  000       LDA A   PFLG,D              ; GET PLUS FLAG.
01D6  80  0F        BSR     FLG1                ; CONDENSE IT.
01D8  96  000       LDA A   MFLG,D              ; GET MINUS FLAG.
01DA  80  0B        BSR     FLG1                ; CONDENSE IT.
01DC  96  000       LDA A   DOLFLG,D            ; GET DOLLAR FLAG.
01DE  80  07        BSR     FLG1                ; CONDENSE IT.
01E0  96  000       LDA A   COMFLG,D            ; GET COMMA FLAG.
01E2  80  03        BSR     FLG1                ; CONDENSE IT.
01E4  D7  000       STA B   FLAGS,D             ; SAVE FOR USAGE.
01E6  39            RTS

01E7  26  02        FLG1:   BNE     B7          ; FLAG IS NOT 0.
01E9  58                    ASL B               ; MAKE ROOM.  BIT = 0.
01EA  39                    RTS                 ; RETURN.
01EB  58            B7:     ASL B               ; MAKE ROOM.
01EC  5C                    INC B               ; BIT = 1.
01ED  39                    RTS                 ; RETURN.


                                        ; FLAGCK CHECKS TO SEE IF ANY ILLEGAL FLAGS (37)
                                        ; ARE SET; IF SO, AN ERROR EXIT IS TAKEN.  IF FLAGS
                                        ; ARE OK, IT CHECKS TO SEE IF NUMFLAG IS SET,
                                        ; IF NOT, IT SETS NUMFLG = TO 1 FOR THE LOOP COUNT.

01EE  8D  03        FLAGCK: BSR     FLAG8       ; CONDENSE FLAGS.
01F0  C5  1F                BIT B   37,I        ; MASK OFF LEGAL FLAGS.
01F2  27  04                BEQ     B17         ; ANY ILLEGAL FLAGS SET?
01F4  33                    PUL B               ; YES, POP RETURN.
01F5  33                    PUL B
01F6  20  46                BRA     MODERR      ; ERROR EXIT.
01F8  D6  000       B17:    LDA B   NUMFLG,D    ; NO, NUMBER SET?
01FA  26  03                BNE     B18         ; YES, EXIT.
01FC  5C                    INC B               ; DEFAULT COUNT = 1.
01FD  D7  000               STA B   NUMFLG,D
01FF  39            B18:    RTS                 ; RETURN
```

```
                                        ; THE FOLLOWING GROUP OF ROUTINES ARE MODIFIER
                                        ; HANDLERS. THEY DEAL WITH ALL MODIFIERS IN THAT
                                        ; IF A MODIFIER IS NOT DUPLICATED, AND IT APPEARS
                                        ; IN THE CORRECT ORDER, IT IS SET; OTHERWISE
                                        ; IT FLAGS THE APPROPRIATE ERROR. OK EXITS
                                        ; ARE MADE TO RECURS. THERE IS ALSO ONE ROUTINE
                                        ; WHICH HANDLES PERIODS (DECIMAL POINTS) AND FLAGS
                                        ; THEM AS ERRORS SINCE A DECIMAL POINT SHOULD
                                        ; NEVER BE SEEN BY THE LEX ANALYZER. THEY ARE
                                        ; HANDLED AS SPECIAL CASES BY 'D' FIELD OPERATORS.

0200  8D  C1      MSR:    BSR     FLAG8           ; CONDENSE FLAGS.
0202  C5  30              BIT B   60,I            ; CHECK FLAGS.
0204  26  38              BNE     MODERR          ; WRONG COMBINATION.
0206  73  0000G           COM     FFLG            ; SET F FLAG.
0209  7E  90E0'    JP:     JMP     RECURS          ; GET NEXT.

020C  8D  B5      PLUS:   BSR     FLAG8           ; CONDENSE FLAGS.
020E  C5  30              BIT B   75,I            ; CHECK FLAGS.
0210  26  2C              BNE     MODERR          ; WRONG COMBINATION.
0212  73  0000G           COM     PFLG            ; SET + FLAG.
0215  20  F2              BRA     JP              ; GET NEXT.

0217  8D  AA      MINUS:  BSR     FLAG8           ; CONDENSE FLAGS.
0219  C5  30              BIT B   75,I            ; CHECK FLAGS.
021B  26  21              BNE     MODERR          ; WRONG COMBINATION.
021D  73  0000G           COM     MFLG            ; SET - FLAG.
0220  20  E7              BRA     JP              ; GET NEXT.

0222  86  00G     PERIOD: LDA A   ERBDPU,I        ; GET PERIOD ERROR.
0224  7E  008C'   JPER:   JMP     EREXIT          ; ERROR EXIT.

0227  8D  9A      DOLLAR: BSR     FLAG8           ; CONDENSE FLAGS.
0229  C5  3F              BIT B   77,I            ; CHECK FLAGS.
022B  26  11              BNE     MODERR          ; WRONG COMBINATION.
022D  86  24              LDA A   '$,I            ; GET ASCII $.
022F  97  00G             STA A   DOLFLG,D        ; SET UP FOR PRINT.
0231  20  D6              BRA     JP              ; GET NEXT.

0233  8D  8E      COMMA:  BSR     FLAG8           ; CONDENSE FLAGS.
0235  C5  31              BIT B   61,I            ; CHECK FLAGS.
0237  26  05              BNE     MODERR          ; WRONG COMBINATION.
0239  73  0000G           COM     COMFLG          ; SET C FLAG.
023C  20  CB              BRA     JP              ; GET NEXT.

023E  86  00G     MODERR: LDA A   EPBMDU,I        ; GET MODIFIER ERROR.
0240  20  E2              BRA     JPER            ; ERROR EXIT.
```

```
                                        ; NOP IS THE COMMA FIELD OERATOR HANDLER.
                                        ; ALL IT DOES IS TO CHECK FOR ANY MODIFIERS
                                        ; PRCEDING THE COMMA.  IF NONE DID, IT RECURSES.
                                        ; IF SOME DID, IT TAKES AN ERROR EXIT.

0242  8D  01C3'     NOP:  JSR    FLAG8      ; CONDENSE FLAGS.
0245  C5  3F              BIT B  77.1       ; ANY MODIFIERS?
0247  27  C0              BEQ    JP         ; NO, GET NEXT OR FINISH.
0249  86  000G      B25N: LDA A  ERBCMU,I    ; NO-OP ERROR.
024B  20  07              BRA    JPER       ; ERROR EXIT.
```

```
                                        ; THE SIMPLE LOOP GROUP HANDLES OUTPUT OF
                                        ; X (SPACES), P (FF'S), L (LF'S), AND / (CR'S).
                                        ; AN ERROR OCCURS IF THE F, +, -, $, OR C FLAG
                                        ; IS SET.  NUMFLG IS AUTOMATICALLY CLEARED BY THE
                                        ; LOOP PROCESS.

024D  86  20      SP:   LDA A  40,I     ; GET SPACES.
024F  97  00G     LOOP: STA A  KO,D     ; SET UP LOOP OUTPUT.
0251  80  98            BSR    FLAGCK   ; CHECK FLAGS.
0253  96  00G     B19:  LDA A  KO,D     ; GET CHARACTER.
0255  BD  0184'         JSR    PUTCHR   ; OUTPUT IT.
0258  7A  00G...        DEC    NUMFLG   ; BUMP DOWN LOOP COUNT.
0258  26  F6            BNE    B19      ; NOT DONE, CONTINUE.
025D  7E  00E0'   SPR:  JMP    RECURS   ; CHECK COMMA, RECURSE EXIT.

0260  86  0C      NP:   LDA A  14,I     ; GET FORM FEED.
0262  20  E8            BRA    LOOP     ; OUTPUT THEM.

0264  86  0A      LF:   LDA A  12,I     ; GET LINE FEED.
0266  20  E7            BRA    LOOP     ; OUTPUT THEM.

0268  86  0D      CR:   LDA A  15,I     ; GET CARRIAGE RETURN.
026A  20  E3            BRA    LOOP     ; OUTPUT THEM.


                                        ; TAB IS USED TO FORCE POSITIONING BY OUTPUTTING
                                        ; THE REQUIRED NUMBER OF FILLER SPACES.  IF THE
                                        ; REQUIRED POSITION HAS ALREADY BEEN PAST, A
                                        ; TAB ERROR (ERTABO) OCCURS.

026C  BD  01EE'   TAB:  JSR    FLAGCK   ; DO PARSE CHECK.
026F  86  20            LDA A  40,I     ; SET UP SPACE LOOP.
0271  97  00G           STA A  KO,D
0273  96  00G           LDA A  NUMFLG,D  ; GET NEXT POSITION.
0275  B0  000GG         SUB A  TABCNT    ; CALCULATE NUMBER OF SPACES.
0278  25  0A            BCS    TABERR    ; ALREADY PAST, DO ERROR.
027A  26  04            BNE    TABON     ; NOT THERE, TAB ON.
027C  97  00G           STA A  NUMFLG,D  ; CLEAR NUMFLG.
027E  20  DD            BRA    SPR       ; NO-OP RETURN.
0280  97  00G     TABON: STA A  NUMFLG,D  ; SET UP SPACES COUNT.
0282  20  CF            BRA    B19       ; GO OUTPUT SPACES.
0284  86  00G     TABERR: LDA A ERTABO,I  ; GET ERROR FLAG.
0286  20  4C            BRA    PEX       ; TAKE ERROR RETURN.
```

```
                                        ; THE S SPECIFIER SETS THE CRLF
                                        ; SUPPRESS FLAG AND MUST BE THE LAST
                                        ; ITEM IN THE IMAGE STRING.


0288  BD  0185'   SCR:   JSR   GETCH     ; LOOK AHEAD
0288  25  06             BCS   LRSERR    ; NOT LAST CHARACTER
028D  86  01             LDA A 1,I       ; GET S FLAG
028F  97  00G            STA A LRSFLG,D  ; SET SUPPRESS FLAG.
0291  20  CA             BRA   SPR       ; FINISH.
0293  86  00G    LRSERR: LDA A ERLRSU,I  ; GET LRS ERROR FLAG
0295  20  3D             BRA   PEX       ; FLAG LRS ERROR
```

```
                                        ; THE REPEAT SPECIFIER (OPEN PARENTHESIS) CHECKS
                                        ; FOR ANY ILLEGAL MODIFIERS AND CHECKS FOR PARENTHESIS
                                        ; OVER-NESTING.  IF EVERYTHING IS OK, IT BUMPS
                                        ; UP THE PAREN COUNT, BUBBLES UP THE PARENTHESIS
                                        ; POINTERS, ADDS NEWEST POINTERS TO STACK, CLEARS NUMFLG, AND
                                        ; EXITS TO RECURS.
                                        ;
                                        ;
                                        ;
0297  BD  01EE'      RS:   JSR   FLAGCK        ; CHECK FLAGS.
029A  BD  0185'            JSR   GETCH        ; ANY CHARACTERS?
029D  25  04               BCS   B22          ; YES, CHECK.
029F  86  00G              LDA A ERINFS,I     ; STRING TOO SHORT.
02A1  20  31               BRA   PEX          ; ERROR EXIT.
02A3  86  04         B22:  LDA A 4,I          ; GET MAX NEST COUNT.
02A5  90  00G              SUB A PARCNT,D     ; COMPARE.
02A7  27  29               BEQ   PER          ; ALREADY AT MAX, ERROR.
02A9  7C  0000G            INC   PARCNT       ; BUMP PARENTHESIS  COUNT.
02AC  DE  00G              LDX   SP3,D        ; BUBBLE UP ADDRESSES.
02AE  DF  00G              STX   SP4,D
02B0  DE  00G              LDX   SP2,D
02B2  DF  00G              STX   SP3,D
02B4  DE  00G              LDX   SP1,D
02B6  DF  00G              STX   SP2,D
02B8  DE  00G              LDX   CURSER,D     ; STORE NEW LOOP ADDRESS.
02BA  DF  00G              STX   SP1,D
02BC  96  00G              LDA A L3,D         ; BUBBLE UP LOOP COUNTS.
02BE  97  00G              STA A L4,D
02C0  96  00G              LDA A L2,D
02C2  97  00G              STA A L3,D
02C4  96  00G              LDA A L1,D
02C6  97  00G              STA A L2,D
02C8  96  00G              LDA A NUMFLG,D     ; STORE NEW LOOP COUNT.
02CA  97  00G              STA A L1,D
02CC  7F  0000G            CLR   NUMFLG       ; CLEAR NUMFLG.
02CF  7E  00E0'      PERJ: JMP   RECURS       ; CONTINUE
                                        ;
02D2  86  00G        PER:  LDA A ERRPNU,I     ; GET PARENTHESIS ERROR.
02D4  7E  008C'      PEX:  JMP   EREXIT       ; ERROR EXIT.
```

```
                                        ; ERF IS THE END REPEAT SPECIFIER AND TAKES
                                        ; CARE OF THE REPEAT COUNTS AND PULLS THE
                                        ; INNER POINTERS OFF THE REPEAT STACK EACH
                                        ; TIME IT ENCOUNTERS A 0 REPEAT COUNT; IF
                                        ; THE REPEAT COUNT IS NOT 0, IT RESETS THE
                                        ; CURSER TO POINT BACK TO THE BEGINNING OF
                                        ; THE PRESENT REPEAT FIELD.  IF THE REPEAT STACK
                                        ; IS EMPTY WHEN ERF IS ENCOUNTERED, THIS IS OBVIOUSLY
                                        ; AN ERROR (TOO MANY CLOSE PARENTHESIS).

02D7  BD  01C3'     ERF:  JSR    FLAG8      ; CONDENSE FLAGS.
02DA  C6  3F              BIT B  77,I       ; CHECK MODIFIERS.
02DC  26  3C              BNE    BMOD       ; NON ALLOWED, ERROR.
02DE  C5  40              BIT B  100,I      ; CHECK PAREN COUNT.
02E0  27  E0              BEQ    PER        ; NO PAREN, ERROR.
02E2  7A  0000G           DEC    L1         ; BUMP DOWN CURRENT COUNT.
02E5  27  06              BEQ    B24        ;  0 MEANS RESET NEXT LEVEL.
02E7  DE  00G             LDX    SP1,D      ; RESET CURSER ADDRESS.
02E9  DF  00G             STX    CURSER,D
02EB  20  E2              BRA    PERJ       ; CONTINUE (GET NEXT).
02ED  7A  0000G     B24:  DEC    PARCNT     ; BUMP DOWN NEST COUNT.
02F0  26  02              BNE    B25        ; RESET TO NEXT LEVEL.
02F2  20  0B              BRA    PERJ       ; NO NEXT LEVEL, RETURN.
02F4  DE  00G       B25:  LDX    SP2,D      ; BUBBLE DOWN ADDRESSES.
02F6  DF  00G             STX    SP1,D
02F8  DE  00G             LDX    SP3,D
02FA  DF  00G             STX    SP2,D
02FC  DE  00G             LDX    SP4,D
02FE  DF  00G             STX    SP3,D
0300  96  00G             LDA A  L2,D       ; BUBBLE DOWN LOOP COUNTS.
0302  97  00G             STA A  L1,D
0304  96  00G             LDA A  L3,D
0306  97  00G             STA A  L2,D
0308  96  00G             LDA A  L4,D
030A  97  00G             STA A  L3,D
030C  20  C1              BRA    PERJ       ; CONTINUE (GET NEXT)

030E  7E  023E'     BMOD: JMP    MODERR     ; MODIFIER ERROR.
```

```
                                        ; LI OUTPUTS A LITERAL STRING THAT IS FOUND
                                        ; WITHIN THE FORMAT STRING.  IT ERRORS FOR ANY
                                        ; IMPROPER MODIFIERS OR IF THE LITERAL IS NOT
                                        ; COMPLETED PROPERLY.  THE NUMBER OF TIMES THE
                                        ; LITERAL IS OUTPUT IS DEPENDENT ON THE VALUE
                                        ; OF THE N MODIFIER.
                                        
0311  BD  01EE'      LI:    JSR   FLAGCK        ; CHECK FLAGS.
0314  DE  00G              LDX   CURSER,D      ; SET UP K4 FOR REPEAT.
0316  DF  00G              STX   K4,D
0318  86  FF               LDA A 377,I         ; SET LITERAL FLAG.
031A  97  00G              STA A LITFLG,D
031C  BD  01A6'    LILOOP: JSR   GETCHI        ; GET CHARACTER.
031F  25  05               BCS   LITRY         ; TRY IT FOR OUTPUT.
0321  86  00G              LDA A ERQUOT,I      ; NO TERMINATOR.
0323  7E  008C'            JMP   EREXIT        ; ERROR EXIT.
0326  81  22       LITRY: CMP A 42,I          ; IS IT QUOTE?
0328  27  05               BEQ   LITCK         ; YES, CHECK END.
032A  BD  0184'    LIOUT: JSR   PUTCHR        ; OUTPUT CHARACTER.
032D  20  ED               BRA   LILOOP        ; CONTINUE LOOP.
032F  BD  0185'    LITCK: JSR   GETCH         ; LOOK AHEAD.
0332  24  09               BCC   LITFIN        ; NO MORE CHARACTERS.
0334  81  22               CMP A 42,I          ; IS SECOND CHAR QUOTE?
0336  26  05               BNE   LITFIN        ; NO, FINISH.
0338  BD  01A6'            JSR   GETCHI        ; YES, TAKE QUOTE.
033B  20  ED               BRA   LIOUT         ; OUTPUT IT.
033D  7A  0000G    LITFIN: DEC   NUMFLG        ; DECREMENT LOOP COUNT.
0340  26  06               BNE   LICONT        ; FORCE NEW LOOP.
0342  7F  0000G            CLR   LITFLG        ; CLEAR LITERAL FLAG.
0345  7E  00E0'            JMP   RECURS        ; FINISHED, GET NEXT.
0348  DE  00G      LICONT: LDX   K4,D          ; RESET CURSER.
034A  DF  00G              STX   CURSER,D
034C  20  CE               BRA   LILOOP        ; OUTPUT LITERAL AGAIN.
```

```
                                        ; CS HANDLES THE CHARACTER STRING DATA
                                        ; TYPES (LITERAL OR STRING VARIABLE).  THE
                                        ; DATA TYPE MUST BE = 2; ONLY NUMBER OR 'F'
                                        ; FLAGS ARE ALLOWED; FOR A NON 'F' OUTPUT,
                                        ; THE LENGTH IS LIMITED TO 255 CHARACTERS
                                        ; WITH ANY DIFFERENCE (N - DL) PADDED
                                        ; WITH SPACES.  IF DL IS GREATER THAN N,
                                        ; A FIELD OVERFLOW IS INDICATED.  THE
                                        ; TEMPORARY LOCATIONS ARE USED AS FOLLOWS:
                                        ; K0, K1 ARE THE RELATIVE 'A' CURSER POINTER.
                                        ; K2, K3 ARE THE LOOP COUNT FOR DATA OUTPUT.
                                        ; K4 IS THE LOOP COUNT FOR SPACES.
                                        ; K5, K6 ARE USED FOR TEMPORARY STORAGE IN
                                        ; THE ABSOLUTE CURSER ADDRESS CALCULATION.

034E  96  00G     CS:     LDA A   FMTFLG,D        ; GET MODE FLAG.
0350  27  05              BEQ     B27             ; OK, CONTINUE.
0352  86  00G             LDA A   ERTMOS,I        ; WRONG MODE
0354  7E  008C'   B27A:   JMP     EREXIT          ; ERROR EXIT.
0357  96  00G     B27:    LDA A   DT,D            ; GET TYPE FLAG.
0359  81  02              CMP A   2,I             ; MUST BE STRING (GI1).
035B  27  04              BEQ     B28A            ; OK, CONTINUE.
035D  86  00G             LDA A   ERDATM,I        ; WRONG TYPE.
035F  20  F3              BRA     B27A            ; ERROR EXIT.
0361  BD  01C3'   B28A:   JSR     FLAG8           ; CONDENSE FLAGS.
0364  C5  0F              BIT B   17,I            ; ANY ILLEGAL MODIFIER.
0366  27  04              BEQ     B28             ; NO, CONTINUE.
0368  86  00G             LDA A   ERBMOU,I        ; YES, BAD MODIFIER.
036A  20  E8              BRA     B27A            ; ERROR EXIT.
036C  DE  00G     B28:    LDX     DL,D            ; GET STRING LENGTH.
036E  DF  00G             STX     K2,D            ; SET UP LOOP COUNT.
0370  C5  10              BIT B   20,I            ; IS F SET.
0372  27  09              BEQ     B29             ; NO, MUST BE N.
0374  DE  00G             LDX     DL,D            ; GET STRING LENGTH.
0376  27  35              BEQ     AFIN            ; NULL STRING, EXIT.
0378  7F  0000G           CLR     K4              ; SPACE COUTN = 0.
037B  20  13              BRA     AOUT            ; OUTPUT STRING.
037D  96  00G     B29:    LDA A   DL,D            ; GET HIGH ORDER BYTE.
037F  27  04              BEQ     B30             ; MUST BE 0, OK, CONTINUE.
0381  86  00G     AFO:    LDA A   ERFLDO,I        ; FIELD OVERFLOW.
0383  20  CF              BRA     B27A            ; ERROR EXIT.
0385  96  00G     B30:    LDA A   NUMFLG,D        ; GET NUMBER.
0387  26  01              BNE     B31             ; NOT 0, USE AS IS.
0389  4C                 INC A                   ; MAKE = 1.
038A  90  01G     B31:    SUB A   DL+1,D          ; COMPARE TO STRING LENGTH
038C  25  F3              BCS     AFO             ; FIELD OVERFLOW.
038E  97  00G             STA A   K4,D            ; SET UP SPACES COUNT.
0390  7F  0000G   AOUT:   CLR     K0              ; SET CURSER.
0393  7F  0000G           CLR     K1
0396  8D  1C      ALOOP:  BSR     AGET            ; GET CHARACTER.
0398  24  05              BCC     SPACES          ; NONE LEFT.
039A  BD  01B4'           JSR     PUTCHR          ; OUTPUT IT.
039D  20  F7              BRA     ALOOP           ; CONTINUE.
039F  96  00G     SPACES: LDA A   K4,D            ; GET SPACES COUNT.
03A1  27  0A      SLP:    BEQ     AFIN            ; FINISHED.
```

```
03A3    86    20              LDA A   40,I         ; A = SPACE.
03A6    BD    0164'           JSR     PUTCHR       ; OUTPUT IT.
03A8    7A    0000G           DEC     K4           ; BUMP DOWN COUNT.
03AB    20    F4              BRA     SLP          ; CONTINUE LOOP.

03AD    4F            AFIN:   CLR A                ; A = 0.
03AE    97    00G             STA A   FFLG,D       ; RESET F FLAG.
03B0    97    00G             STA A   NUMFLG,D     ; RESET NUMFLG.
03B2    32                    PUL A                ; SCRATCH TAG.
03B3    39                    RTS                  ; RETURN TO CALLER.

03B4    DE    00G     AGET:   LDX     K2,D         ; GET COUNT.
03B6    26    02              BNE     B32          ; SOME LEFT, CONTINUE.
03B8    0C                    CLC                  ; C = 0 = NONE LEFT.
03B9    39                    RTS                  ; RETURN.
03BA    09            B32:    DEX                  ; BUMP DOWN LOOP COUNT.
03BB    DF    00G             STX     K2,D         ; SAVE IT.
03BD    96    00G             LDA A   K0,D         ; MAKE CURSER RESOLUTE
03BF    D6    00G             LDA B   K1,D
03C1    DB    01G             ADD B   DP+1,D
03C3    99    00G             ADC A   DP,D
03C5    97    00G             STA A   K5,D         ; SAVE IT.
03C7    D7    00G             STA B   K6,D
03C9    DE    00G             LDX     K5,D         ; SET X = CHARACTER POINTER
03CB    A6    00              LDA A   0,X          ; GET CHARACTER.
03CD    0D                    SEC                  ; C = 1.
03CE    DE    00G             LDX     K0,D         ; GET CURSER
03D0    08                    INX                  ; INCREMENT IT.
03D1    DF    00G             STX     K0,D         ; SAVE IT.
03D3    39                    RTS                  ; RETURN.
```

```
                                        ; SF OUTPUTS A NUMERIC DATA TYPE IN SCIENTIFIC FORMAT.
                                        ; THE NUMBER IS FORMATTED ACCORDING TO THE MODIFIERS
                                        ; BEFORE BEING OUTPUT.  SF WILL ERROR EXIT FOR THE
                                        ; FOLLOWING CONDITIONS:
                                        ;    1) WRONG MODE
                                        ;    2) DATA TYPE MISMATCH
                                        ;    3) ILLEGAL MODIFIERS
                                        ;    4) NUMBER OUT OF RANGE

03D4  96 00G        SF:   LDA A  FMTFLG,D    ; GET MODE FLAG.
03D6  27 05               BEQ    B34         ; OK, CONTINUE.
03D8  86 00G              LDA A  ERTMDS,I    ; WRONG MODE.
03DA  7E 008C'      B33:  JMP    EREXIT      ; ERROR EXIT.
03DD  96 00G        B34:  LDA A  DT,D        ; GET TYPE FLAG.
03DF  81 01               CMP A  1,I         ; MUST BE NUMERIC.
03E1  27 04               BEQ    B35         ; OK, CONTINUE.
03E3  86 00G              LDA A  ERDATM,I    ; WRONG TYPE.
03E5  20 F3               BRA    B33         ; ERROR EXIT.
03E7  BD 01C3'      B35:  JSR    FLAGS       ; CONDENSE FLAGS.
03EA  C5 07               BIT B  7,I         ; ANY ILLEGAL MODIFIERS.
03EC  27 04               BEQ    B36         ; NO, CONTINUE.
03EE  86 00G              LDA A  ERBMOD,I    ; YES, GET ERROR FLAG.
03F0  20 E8               BRA    B33         ; ERROR EXIT.
03F2  96 00G        B36:  LDA A  FFLG,D      ; GET AS REQUIRED FLAG.
03F4  27 0E               BEQ    NE          ; NOT SET, MUST BE NUMBER.
03F6  86 0A               LDA A  10.,I       ; GET MAX COUNT.
03F8  BD 048F'            JSR    ROUND       ; ROUND THE NUMBER.

03FB  16                  TAB                ; SAVE A, B=LOOP COUNT.
03FC  81 02               CMP A  2,I         ; COUNT EQ OR GT 2?
03FE  24 25               BCC    EOUT        ; YES, OUTPUT NUMBER.
0400  C6 02               LDA B  2,I         ; NO, SET DEFAULT MIN=2.
0402  20 21               BRA    EOUT        ; NOW OUTPUT MIN #.
0404  D6 00G        NE:   LDA B  NUMFLG,D    ; GET NUMBER
0406  26 09               BNE    NMCK        ; SET, CHECK RANGE.
0408  86 02               LDA A  2,I         ; NOT SET, GET MINIMUM.
040A  BD 048F'            JSR    ROUND       ; SET UP FOR OUTPUT.
040D  C6 02               LDA B  2,I         ; B=LOOP COUNT.
040F  20 14               BRA    EOUT        ; OUTPUT NUMBER.
0411  7E 0000G      NMCK: CLR    NUMFLG      ; CLEAR FLAG
0414  86 0B               LDA A  11.,I       ; GET MAX DIGITS.
0416  11                  CBA                ; COMPARE TO MAX
0417  24 04               BCC    NECONT      ; OK RANGE (1-11).
0419  86 00G              LDA A  ERINU,I     ; TOO MANY DIGITS.
041B  20 BD               BRA    B33
041D  5C            NECONT: INC B            ; BUMP FOR ROUND OFF.
041E  D7 00G              STA B  K6,D        ; SAVE POSITIONS.
0420  17                  TBA                ; A=ROUND POINTER.
0421  8D 6C               BSR    ROUND       ; SET FOR OUTPUT.
0423  D6 00G              LDA B  K5,D        ; B=OUTPUT LOOP COUNT.
0425  7F 0000G      EOUT: CLR    PFLG        ; CLEAR FLAGS.
0428  7E 0000G            CLR    FFLG
042B  D7 00G              STA B  K6,D
042D  B6 0000G            LDA A  NS          ; OUTPUT SIGNS
0430  81 2D               CMP A  '-,I        ; SIGN = - ?
0432  27 05               BEQ    EOUTS       ; YES, OUTPUT IT.
```

```
                                                LDA B   DPCONT          ; NO, IN DEFAULT MODE?
                                                BNE     ETMSD           ; YES, SKIP SIGN OUTPUT.
                                EOUTS:  JSR     PUTCHR
                                ETMSD:  LDA A   N12             ; OUTPUT MSD.
                                        JSR     PUTCHR
                                        LDA A   '.,I            ; OUTPUT PERIOD.
                                        JSR     PUTCHR
                                        DEC     K6              ; BUMP DOWN LOOP COUNT.
                                        LDX     N11,I           ; X=DIGIT POINTER.
                                EO:     STX     K0,D            ; SAVE ADDRESS.
                                        LDA A   0,X             ; GET DIGIT.
                                        JSR     PUTCHR          ; OUTPUT IT.
                                        DEC     K6              ; BUMP DOWN LOOP COUNT.
                                        BEQ     ECONT           ; FINISHED, OUTPUT EXPONENT.
                                        LDX     K0,D            ; GET POINTER.
                                        DEX                     ; POINT TO NEXT DIGIT.
                                        BRA     EOLP            ; CONTINUE LOOP.
                                ECONT:  LDA A   'E,I            ; OUTPUT E.
                                        JSR     PUTCHR
                                        LDA A   ES              ; OUTPUT SIGN.
                                        JSR     PUTCHR
                                        LDA A   E3              ; OUTPUT E MSD.
                                        LDA B   DPCONT          ; GET DEFAULT FLAG.
                                        BEQ     ECONT3          ; NOT DEFAULT, PRINT ALL THREE.
                                        CMP A   '0,I            ; FIRST DIGIT 0?
                                        BNE     ECONT3          ; NO, PRINT ALL THREE.
                                        LDA A   E2              ; GET SECOND DIGIT.
                                        CMP A   '0,I            ; 0?
                                        BNE     ECONT2          ; NO, PRINT 2 DIGITS.
                                        BRA     ECONT1          ; PRINT LAST DIGIT ONLY.
                                ECONT3: JSR     PUTCHR
                                        LDA A   E2              ; OUTPUT NEXT DIGIT.
                                ECONT2: JSR     PUTCHR
                                ECONT1: LDA A   E1              ; OUTPUT LSD.
                                        JSR     PUTCHR
                                        PUL A                   ; POP TAG.
                                        RTS                     ; RETURN TO CALLER.
```

```
                                        ; ROUND PREPARES THE DECIMAL BCD STRING IN
                                        ; E1-N12 FOR OUTPUT.  ON ENTRY 'A' MUST CONTAIN
                                        ; THE NUMBER OF CHARACTERS REQUIRED IN THE FINAL
                                        ; ANSWER (1-12).  ON EXIT, 'A' CONTAINS THE NUMBER
                                        ; OF DIGITS TO THE LEFT OF ANY TRAILING ZEROS (0-12).
                                        ; IN PARTICULAR, ROUND DOES THE FOLLOWING:
                                        ;   1) ROUNDS THE NUMBER.
                                        ;   2) CONVERTS THE STRING FROM BCD TO ASCII.
                                        ;   3) ENSURES THE SIGNS ARE PROPER WITH RESPECT TO
                                        ;      THE NUMBER VALUE AND THE + MODIFIER.
                                        ;   4) CALCULATES THE NUMBER OF DIGITS TO THE LEFT
                                        ;      OF ANY TRAILING ZEROS.
                                        ;
                                        ;
049F F6 0000G       ROUND:  LDA B  N12          ; GET 0 FLAG.
04A2 26 21                  BNE    RNDN         ; NUMBER NOT 0, ROUND IT.
04A4 D7 00G                 STA B  EXH,D        ; CLEAR BINARY EXPONENT.
04A6 D7 00G                 STA B  EXL,D
04A8 F7 0000G               STA B  DIGCNT       ; DIGIT COUNT = 0.
04AB 86 0F                  LDA A  15.,I        ; SET UP LOOP.
04AD C6 30                  LDA B  '0,I
04AF CE 0000G               LDX    E1,I
04B2 E7 00          RNDLP:  STA B  0,X          ; STORE ASCII 0.
04B4 4A                     DEC A               ; BUMP DOWN LOOP COUNT.
04B5 27 03                  BEQ    RNDO         ; 0= FINISHED LOOP.
04B7 08                     INX                 ; BUMP UP ADDRESS.
04B8 20 F8                  BRA    RNDLP        ; CONTINUE LOOP.
04BA C6 20          RNDO:   LDA B  40,I         ; SIGNS=SPACE=+.
04BC F7 0000G               STA B  NS
04BF C6 2B                  LDA B  '+,I
04C1 F7 0000G               STA B  ES
04C4 39             RNDRET: RTS                 ; RETURN.
                                        ;
04C5 BD 0000G       RNDN:   JSR    RNDATA       ; ROUND NUMBER.
04C8 B7 0000G               STA A  DIGCNT       ; SAVE DIGITS COUNT.
04CB 86 0C                  LDA A  12.,I
04CD B0 0000G               SUB A  DIGCNT
04D0 27 10                  BEQ    RSN
04D2 CE 0000G               LDX    N1,I
04D5 C6 30                  LDA B  60,I
04D7 E7 00          RSNLP:  STA B  0,X
04D9 4A                     DEC A
04DA 27 03                  BEQ    RSNO
04DC 08                     INX
04DD 20 F8                  BRA    RSNLP
04DF B6 0000G       RSNO:   LDA A  DIGCNT
04E2 D6 00G         RSN:    LDA B  PFLG,D       ; GET PLUS FLAG.
04E4 27 0C                  BEQ    RSNR         ; NOT SET, RETURN.
04E6 F6 0000G               LDA B  NS           ; GET NUMBER SIGN.
04E9 C1 20                  CMP B  40,I         ; IS IT POSITIVE?
04EB 26 05                  BNE    RSNR         ; NO, RETURN.
04ED C6 2B                  LDA B  '+,I         ; YES, MAKE +.
04EF F7 0000G               STA B  NS
04F2 39             RSNR:   RTS                 ; RETURN.
```

```
                                        ; D1 IS USED TO HANDLE THE 'D' OPERATOR: IT IS
                                        ; DIVIDED INTO THREE MAIN SECTION: SYNTAX ANALYSIS,
                                        ; SET UP PRINT FIELDS, AND OUTPUT PRINT FIELDS.  ITS
                                        ; FOUR MAIN ERROR TYPES ARE WRONG MODE, SYNTAX
                                        ; ERRORS, FIELD OVERFLOWS, AND MODIFIERS ERRORS.
                                        ; IT SHARES THE ROUND ROUTINE WITH SF.

 10  04E3  96  000  D1:    LDA A  FMTFLG,D    ; GET MODE FLAG.
 11  04E5  27  05   B141    BEQ    B141        ; IN FMTOUT MODE OK.
 12  04E7  86  000          LDA A  ERTMDS,I    ; WRONG MODE.
 13  04E9  7E  008C' B140:  JMP    EREXIT      ; ERROR EXIT.
 14  04EC  96  000  B141:   LDA A  DT,D        ; GET DATA TYPE
 15  04EE  81  01           CMP A  1,I         ; IS IT NUMERIC?
 16  04F0  27  04           BEQ    B142        ; YES, CONTINUE.
 17  04F2  86  000          LDA A  ERDATM,I    ; NO, DATA TYPE MISMATCH.
 18  04F4  20  F3           BRA    B140        ; ERROR EXIT.
 19  04F6  4F               CLR H              ; A=0.
 20  04F7  B7  0000G        STA A  DDP         ; CLEAR DECIMAL POINT FLAG
 21  04FA  B7  0000G        STA A  DIGCNT      ; CLEAR DIGIT COUNT.
 22  04FD  B7  0000G        STA A  DN1         ; CLEAR COUNTS.
 23  0500  B7  0000G        STA A  DN2
 24  0503  B7  0000G        STA A  F1          ; CLEAR AS REQUIREDS.
 25  0506  B7  0000G        STA A  F2
 26  0509  96  000          LDA A  NUMFLG,D    ; GET NUMBER
 27  050B  27  08           BEQ    DEFCK       ; CHECK DEFAULT VALUE.
 28  050D  B7  0000G        STA A  DN1         ; SAVE IT.
 29  0510  7F  0000G        CLR    NUMFLG      ; CLEAR FLAG FOR NEXT.
 30  0513  20  11           BRA    SYNTX1      ; DO D SYNTAX.
 31  0515  96  000  DEFCK:  LDA A  FFLG,D       ; GET F FLAG.
 32  0517  27  08           BEQ    DEFALT      ; NOT SET, DO DEFAULT
 33  0519  B7  0000G        STA A  F1          ; SET, SET UP LEFT AS REQ.
 34  051C  7F  0000G        CLR    FFLG        ; CLEAR F FLAG.
 35  051F  20  05           BRA    SYNTX1      ; FINISH SYNTAX CHECK.
 36  0521  86  01   DEFALT: LDA A  1,I         ; DEFAULT=1.
 37  0523  B7  0000G        STA A  DN1
 38  0526  BD  0185' SYNTX1: JSR    GETCH       ; LOOK AHEAD.
 39  0529  24  63           BCC    DEND        ; END OF STRING.
 40  052B  81  2E           CMP A  '.,I        ; = DECIMAL POINT?
11111
 41  052D  26  5F           BNE    DEND        ; NO, END OF D INPUT.
 42  052F  B7  0000G        STA A  DDP
 43  0532  BD  01AE'        JSR    INCCUR      ; TAKE DP.
 44  0535  BD  0185'        JSR    GETCH       ; LOOK AHEAD.
 45  0538  24  54           BCC    DEND        ; END OF STRING.
 46  053A  16               TAB                ; SAVE CHAR.
 47  053B  C1  44           CMP B  'D,I        ; =D?
 48  053D  26  0A           BNE    SYNTX2      ; NO, CHECK FOR NUMBER/F.
 49  053F  86  01           LDA A  1,I         ; YES, DEFAULT=1.
 50  0541  B7  0000G        STA A  DN2
 51  0544  BD  01AE'        JSR    INCCUR      ; TAKE CHARACTER.
 52  0547  20  45           BRA    DEND        ; END OF D INPUT.
 53  0549  BD  0177' SYNTX2: JSR    CKNUM       ; IS IT NUMBER.
 54  054C  24  1D           BCC    CHKF        ; NO.
 55  054E  BD  0186'        JSR    GETCH1      ; TAKE FIRST CHARACTER.
 56  0551  BD  0177'        JSR    CKNUM       ; MAKE BCD.
 57  0554  BD  0123'        JSR    GETNUM      ; YES, TAKE IT.
```

```
0557  BD  0185'          JSR    GETCH       ; LOOK AHEAD.
055A  81    44           CMP A  'D,I        ; DO WE HAVE D?
055C  26    30           BNE    DEND        ; NO, LEAVE # ALONE.
055E  BD  01AE'          JSR    INCCUR      ; YES, EAT IT.
0561  96   00G           LDA A  NUMFLG,D    ; TAKE NUMBER.
0563  B7  0000G          STA A  DN2         ; SAVE IT.
0566  7F  0000G          CLR    NUMFLG      ; CLEAR THE FLAG.
0569  20    23           BRA    DEND
056B  81    46   CHKF:   CMP A  'F,I        ; IS IT F?
056D  26    1F           BNE    DEND        ; NO, END OF D INPUT.
056F  BD  01AE'          JSR    INCCUR      ; YES, TAKE IT.
0572  BD  0185'          JSR    GETCH       ; LOOK AHEAD
0575  25    05           BCS    CHKF2       ; WE HAVE ANOTHER CHAR.
0577  86   00G           LDA A  ERRNOU,I    ; END OF STRING, ERROR
0579  7E  008C'          JMP    EREXIT      ; ERROR EXIT.
057C  81    44   CHKF2:  CMP A  'D,I        ; IS IT D?
057E  27    06           BEQ    SETF2       ; YES, SET F2
0580  86    FF           LDA A  377,I       ; NO, SET F FLAG.
0582  97   00G           STA A  FFLG,D
0584  20    08           BRA    DEND        ; END OF D INPUT
0586  BD  01AE'  SETF2:  JSR    INCCUR      ; TAKE D.
0589  86    FF           LDA A  377,I       ; SET LEFT AS REQUIRED
058B  B7  0000G          STA A  F2
```

```
                                                        ; THIS SECTION DOES ROUNDING
                                                        ; FOR D OPERATOR.
058E  B6 0000G     DEND:  LDA A  F2        ; GET DECIMAL "AS REQUIRED" FLAG.
0591  27 0D                BEQ    DON2      ; NOT SET, DO NUMBER.
0593  86 0C                LDA A  12.,1     ; ROUND AT MAXIMUM.
0595  BD 048F'             JSR    ROUND
0598  BD 02EB'             JSR    CONEXP    ; CONVERT AND CHECK EXPONENT.
0598  24 09                BCC    TF2       ; NO ERRORS.
059D  7E 0818'             JMP    LER14     ; EXPONENT ERROR.
05A0  B6 0000G     DON2:  LDA A  DN2       ; GET DECIMAL ROUND POINTER.
05A3  BD 079F'             JSR    RNDATN    ; ROUND AT N.
                                                        ; HERE WE BEGIN TO SET UP
                                                        ; PRINT FIELDS.
05A6  BD 0773'     TF2:   JSR    SETDF     ; SET UP FLAG DIRECTED OUTPUT.
05A9  BD 0820'             JSR    CALCOM    ; CALCULATE COMMA COUNT.

05AC  B6 0000G             LDA A  DIGCNT    ; GET PRINTABLE DIGITS.
05AF  26 0E                BNE    NZ        ; NOT ZERO, GO FORMAT.
05B1  86 01        SETZ:   LDA A  1,I       ; COUNT=1
05B3  B7 0000G             STA A  LDIG
05B6  CE 05BE'             LDX    A0,I      ; POINT TO ASCII 0.
05B9  FF 0000G             STX    LDIGA
05BC  20 26                BRA    CKFORN    ; CHECK N COUNT.

05BE  30           A0:     BYTE   60        ; ASCII 0.

05BF  B6 0000G     NZ:    LDA A  EXS        ; GET SIGN.
05C2  26 ED                BNE    SETZ      ; MINUS, SET UP LEFT "PRINT 0".

05C4  7C 0000G             INC    EXP       ; MOVE DECIMAL POINT LEFT.

05C7  80 4E        B60:    BSR    EMD       ; EXP-DIGCNT.

05C9  2B 00                BMI    B61       ; DO "GREATER" CASE.

05CB  B6 0000G             LDA A  DIGCNT    ; SET UP LEFT DIGIT COUNT.
05CE  B7 0000G             STA A  LDIG
05D1  96 00G               LDA A  K0,D      ; SET UP LEFT TRAILING 0 COUNT.
05D3  B7 0000G             STA A  ZERO1
05D6  20 06                BRA    B62       ; SET UP ADDRESS.

05D8  B6 0000G     B61:    LDA A  EXP       ; DIGIT PRINT COUNT=EXP.
05DB  B7 0000G             STA A  LDIG
05DE  CE 0000G     B62:    LDX    N12,I     ; GET DIGITS ADDRESS.
05E1  FF 0000G             STX    LDIGA     ; SET UP FOR PRINT POINTER.

05E4  B6 0000G     CKFORN: LDA A  F1        ; GET LEFT "AS REQUIRED" FLAG.
05E7  26 37                BNE    CKDP      ; SET, SKIP LIMIT CHECKS.

05E9  B6 0000G             LDA A  LDIG      ; GET LEFT DIGIT COUNT.
05EC  F6 0000G             LDA B  ZERO1     ; GET TRAILING 0 COUNT.
05EF  1B                   ABA              ; ACCUMULATE SUM.
05F0  F6 0000G             LDA B  COMCNT    ; GET COMMAS COUNT.
05F3  1B                   ABA              ; ADD IN.
05F4  97 00G               STA A  K0,D      ; SAVE SUM.
05F6  96 00G               LDA A  MFLG,D    ; GET MINUS FLAG
```

```
                05F8  26  0C              BNE   FINCK          ; FINISH CHECK.
                05FA  96  00G             LDA A PELG.D         ; GET PLUS FLAG.
                05FC  26  08              BNE   FINCK          ; FINISH CHECK.
                05FE  B6  0000G           LDA A NS             ; GET SIGN.
                0601  27  03              BEQ   FINCK          ; FINISH CHECK.
                0603  7C  0000G           INC   KO             ; COUNT SIGN IN TOTAL.
                
                0606  B6  0000G   FINCK:  LDA A DN1            ; DN1-TOTAL.
                0609  90  00G             SUB A KO,D
                0608  24  05              BCC   B63            ; POSITIVE RESULT=OK.
                0600  86  00G             LDA A EREL00.1       ; GET FIELD OVERFLOW.
                060F  7E  008C'           JMP   EREXIT         ; TAKE ERROR EXIT.
                0612  B7  0000G   B63:    STA A LSPS           ; SET UP LEADING SPACES.
                0615  20  09              BRA   CKDP           ; GO DO RIGHT FIELD.
                
                
                0617  B6  0000G   END:    LDA A EXP            ; EXP-DIGCNT.
                061A  B0  0000G           SUB A DIGCNT
                061D  97  00G             STA A KO,D
                061F  39                  RTS                  ; RETURN
```

```
                                                        ; BEGIN SET-UP OF RIGHT DIGIT FIELD HERE.
                                                        ;
0620  BD 07FB'    CKDP:  JSR     CONEXP     ; RESTORE EXPONENT.
0623  24 03               BCC     BX1        ; NO ERRORS.
0625  7E 0818'            JMP     LER14      ; EXPONENT TOO LARGE.
0628  B6 0000G    BX1:   LDA A   DIGCNT     ; GET NUMBER CONTROL.
062B  26 0F               BNE     B65        ; HAVE SOMETHING. SET IT UP.
062D  B6 0000G            LDA A   F2         ; GET RIGHT "AS REQUIRED" FLAG.
0630  27 02               BEQ     B64        ; NOT SET, CHECK RIGHT N MOD.
0632  20 50               BRA     DIGOUT     ; OUTPUT NUMBER.
0634  B6 0000G    B64:   LDA A   DN2        ; GET NUMBER.
0637  B7 0000G            STA A   ZERO2      ; SET UP RIGHT 0 PRINT COUNT.
063A  20 55               BRA     DIGOUT     ; OUTPUT NUMBER.
                                                        ;
063C  B6 0000G    B65:   LDA A   EXS        ; GET SIGN
063F  26 26               BNE     B66        ; NEGATIVE.CHECK FOR POSITION.
                                                        ;
                                                        ;
0641  7C 0000G            INC     EXP        ; MAKE EXPONENT ABSOLUTE.
0644  8D D1               BSR     END        ; EXP-DIGCNT.
0646  24 EC               BCC     B64        ; NO DECIMAL DIGITS.
0648  40                  NEG A              ; MAKE POSITIVE.
0649  B7 0000G            STA A   RDIG       ; SET UP RIGHT DIGIT COUNT.
064C  97 00G              STA A   KO.D       ; FLIP SIGN OF RESULT.
064E  F6 0000G            LDA B   EXP        ; SET LOOP COUNT.
0651  CE 0000G            LDX     N12.1      ; GET STARTING ADDRESS.
0654  09          B67:    DEX                ; BUMP DOWN ADDRESS.
0655  5A                  DEC B              ; BUMP DOWN LOOP COUNT.
0656  26 FC               BNE     B67        ; CONTINUE LOOP.
0658  FF 0000G            STX     RDIGA      ; SET UP PRINT ADDRESS.
065B  F6 0000G            LDA B   DN2        ; GET FIELD SIZE.
065E  27 31               BEQ     DIGOUT     ; NOT REQUIRED, GO OUTPUT.
0660  D0 00G              SUB B   KO.D       ; CALCUALTE TRAILING ZEROS.
0662  F7 0000G            STA B   ZERO3      ; SAVE RESULT FOR OUTPUT.
0665  20 2A               BRA     DIGOUT     ; GO OUTPUT DIGITS.
                                                        ;
                                                        ;
0667  B6 0000G    B66:   LDA A   EXP        ; GET EXPONENT.
066A  4A                  DEC A              ; MAKE ABSOLUTE.
066B  B7 0000G            STA A   ZERO2      ; SET UP LEADING ZEROS.
066E  B6 0000G            LDA A   DIGCNT     ; GETDIGIT COUNT.
0671  B7 0000G            STA A   RDIG       ; SET UP FOR PRINT.
0674  CE 0000G            LDX     N12.1      ; GET DIGIT ADDRESS.
0677  FF 0000G            STX     RDIGA      ; SET UP FOR PRINT.
067A  B6 0000G            LDA A   F2         ; GET "AS REQUIRED" FLAG.
067D  26 12               BNE     DIGOUT     ; SET, NO TRAILING ZEROS.
067F  B6 0000G            LDA A   DN2        ; GET NUMBER.
0682  27 0D               BEQ     DIGOUT     ; NO TRAILING ZEROS REQUIRED.
0684  B0 0000G            SUB A   ZERO2      ; SUBTRACT OU PRINT MATERIAL.
0687  2B 08               BMI     DIGOUT
0689  B0 0000G            SUB A   DIGCNT
068C  2B 03               BMI     DIGOUT
068E  B7 0000G            STA A   ZERO3      ; SET UP ANY TRAILING 0'S.
```

```
                                                        ; START ACTUAL OUTPUT HERE.

0691  B6 0000G   DIGOUT: LDA A LSPS      ; GET LEADING SPARE COUNT.
0694  27 0A              BEQ   DOLOUT     ; NONE REQUIRED.
0696  B6 20      B41:    LDA A #0.1      ; GET SPACE.
0698  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
069B  7A 0000G           DEC   LSPS       ; BUMP DOWN LOOP COUNT.
069E  26 F6              BNE   B41        ; CONTINUE LOOP.
06A0  96 00G     DOLOUT: LDA A DOLFLG,D   ; GET DOLLAR SIGN PRINT.
06A2  27 03              BEQ   SGNOUT     ; NONE REQUIRED.
06A4  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
06A7  B6 0000G   SGNOUT: LDA A NS         ; GET SIGN.
06AA  27 03              BEQ   LDIGCK     ; NONE REQUIRED.
06AC  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
06AF  B6 0000G   LDIGCK: LDA A LCOM       ; GET COMMA REMAINDER.
06B2  26 1E              BNE   COMOUT     ; OUTPUT # WITH COMMAS.
06B4  FE 0000G           LDX   LDIGA      ; GET DIGIT ADDRESS.
06B7  A6 00      B42:    LDA A 0,X        ; GET CHARACTER.
06B9  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
06BC  7A 0000G           DEC   LDIG       ; DECREMENT COUNT.
06BF  27 09              BEQ   B43        ; 0=FINISHED DIGITS.
06C1  FE 0000G           LDX   LDIGA      ; CALCULATE NEXT ADDRESS.
06C4  09                 DEX
06C5  FF 0000G           STX   LDIGA      ; SAVE RESULT.
06C8  20 ED              BRA   B42        ; CONTINUE LOOP.
06CA  B6 0000G   B43:    LDA A ZERO1      ; GET TRAILING ZERO COUNT.
06CD  BD 0756'           JSR   ZEROUT     ; OUTPUT ANY TRAILING 0'S.
06D0  20 47              BRA   DPOUT      ; OUTPUT DECIMAL POINT.

06D2  FE 0000G   COMOUT: LDX   LDIGA      ; GET DIGIT ADDRESS.
06D5  A6 00              LDA A 0,X        ; GET DIGIT FOR OUTPUT.
06D7  09                 DEX              ; CALCULATE NEXT DIGIT.
06D8  FF 0000G           STX   LDIGA      ; SAVE RESULT.
06DB  BD 0184'           JSR   PUTCHR     ; OUTPUT CHARACTER.
06DE  7A 0000G           DEC   LDIG       ; BUMP DOWN DIGIT COUNT.
06E1  27 11              BEQ   ZRCK       ; 0=NO MORE DIGITS.
06E3  7A 0000G           DEC   LCOM       ; BUMP DOWN COMMA COUNT.
06E6  26 EA              BNE   COMOUT     ; CONTINUE LOOP.
06E8  86 2C              LDA A #',1       ; GET COMMA.
06EA  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
06ED  86 03              LDA A #3,1       ; RESET LCOM.
06EF  B7 0000G           STA A LCOM
06F2  20 DE              BRA   COMOUT     ; CONTINUE LOOP.

06F4  B6 0000G   ZRCK:   LDA A ZERO1      ; GET ZERO COUNT.
06F7  27 20              BEQ   DPOUT      ; NONE REQUIRED.
06F9  7A 0000G           DEC   LCOM       ; PLAY CATCH-UP.
06FC  27 0F              BEQ   B44C       ; NEEDS RESET.
06FE  86 30      B44:    LDA A #'0,1      ; GET ASCII 0.
0700  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
0703  7A 0000G           DEC   ZERO1      ; BUMP DOWN LOOP COUNT.
0706  27 11              BEQ   DPOUT      ; 0=FINISHED.
0708  7A 0000G           DEC   LCOM       ; BUMP DOWN COMMA COUNT.
070B  26 F1              BNE   B44        ; CONTINUE LOOP.

070D  86 2C      B44C:   LDA A #',1       ; GET COMMA.
070F  BD 0184'           JSR   PUTCHR     ; OUTPUT IT.
```

```
0712  86  03                LDA A  3,I      ; RESET COMMA COUNT.
0714  B7  0000G             STA A  LCOM
0717  20  E5                BRA    B44      ; CONTINUE LOOP.

0719  B6  0000G    DPOUT:   LDA A  DOP      ; GET DECIMAL POINT PRINT.
071C  27  20                BEQ    DEXIT    ; NOT REQUIRED, DO CLEAN-UP.

071E  F6  0000G    COUTDP:  LDA B  DPCONT   ; GET DECIMAL.
0721  27  05                BEQ    OUTIT    ; 0=FMTOUT MOD, OUTPUT IT.
0723  F6  0000G             LDA B  RDIG     ; CHECK FOR ANY RIGHT DIGITS.
0726  27  23                BEQ    DEXIT    ; NONE, FINISHED.
0728  BD  0184'    OUTIT:   JSR    PUTCHR   ; OUTPUT DECIMAL POINT
072B  B6  0000G             LDA A  ZERO2    ; OUTPUT ANY LEADING 0'S.
072E  8D  26                BSR    ZEROUT
0730  B6  0000G             LDA A  RDIG     ; GET DIGIT COUNT.
0733  27  16                BEQ    DEXIT    ; NONE REQUIRED.
0735  FE  0000G    DZN:     LDX    RDIGA    ; GET DIGITS ADDRESS
0738  A6  00                LDA A  0,X      ; GET DIGIT
073A  09                    DEX             ; CALCULATE NEXT ADDRESS.
073B  FF  0000G             STX    RDIGA    ; SAVE IT.
073E  BD  0184'             JSR    PUTCHR   ; OUTPUT CHARACTER.
0741  7A  0000G             DEC    RDIG     ; BUMP DOWN COUNT.
0744  26  EF                BNE    DZN      ; CONTINUE LOOP.
0746  B6  0000G             LDA A  ZERO3    ; GET TRAILING ZERO COUNT.
0749  8D  0B                BSR    ZEROUT   ; OUTPUT ANY TRAILING 0'S.

074B  4F           DEXIT:   CLR A           ; CLEAR MODIFIER FLAGS.
074C  97  00G               STA A  PFLG,D
074E  97  00G               STA A  MFLG,D
0750  97  00G               STA A  DOLFLG,D
0752  97  00G               STA A  COMFLG,D
0754  32                    PUL A           ; POP TAG.
0755  39                    RTS             ; RETURN.

0756  26  01       ZEROUT:  BNE    Z0       ; ANY ZEROS NEED OUTPUTTING.
0758  39                    RTS             ; NO, RETURN.
0759  B7  0000G    Z0:      STA A  ZERCNT   ; YES, SAVE COUNT.
075C  32                    PUL A           ; POP RETURN.
075D  B7  0000G             STA A  ZERSVH
0760  32                    PUL A
0761  B7  0000G             STA A  ZERSVL
0764  86  30       ZERLP:   LDA A  '0,I     ; GET ASCII 0.
0766  BD  0184'             JSR    PUTCHR   ; OUTPUT IT.
0769  7A  0000G             DEC    ZERCNT   ; BUMP DOWN LOOP COUNT.
076C  26  F6                BNE    ZERLP    ; NOT FINISHED.
076E  FE  0000G             LDX    ZERSVH   ; FINISHED, GET RETURN.
0771  6E  00                JMP    0,X      ; RETURN.
```

```
                                        ; SETDF CLEARS THE LOOPING DIGIT PRINT FIELDS SO IF
                                        ; THEY ARE NOT NEEDED. THEY ARE AUTOMATICALLY SKIPPED
                                        ; IT ALSO SETS UP THE FIXED SIGN CHARACTERS
                                        ; UNDER CONTROL OF THE MODIFIER SPECIFICATIONS.
                                        
                                        
0773  4F          SETDF:  CLR A         ; A=0
0774  B7  0000G           STA A  LSPS   ; CLEAR ALL PRINT CONTROLS.
0777  B7  0000G           STA A  LDIG
077A  B7  0000G           STA A  LCOM
077D  B7  0000G           STA A  COMCNT
0780  B7  0000G           STA A  RDIG
0783  B7  0000G           STA A  ZERO1
0786  B7  0000G           STA A  ZERO2
0789  B7  0000G           STA A  ZERO3
078C  96  00G             LDA A  PFLG,D ; GET PLUS FLAG.
078E  26  0E              BNE    SETOK  ; SET, LEAVE SIGN ALONE.
0790  96  00G             LDA A  MFLG,D ; GET MINUS FLAG.
0792  26  09              BNE    SETOK  ; SET, LEAVE SIGN ALONE.
0794  B6  0000G           LDA A  NS     ; GET SIGN.
0797  81  2D              CMP A  '-,I   ; MINUS?
0799  27  03              BEQ    SETOK  ; YES, LEAVE SIGN ALONE.
079B  7F  0000G           CLR    NS     ; CLEAR PREVIOUS SIGN.
079E  39            SETOK: RTS          ; RETURN.
```

```
                                        ; RNDATN ROUNDS THE NUMBER AT THE DIGIT POINTED
                                        ; TO BY A.
                                        ; ON EXIT, DIGCNT CONTAINS THE COUNT OF THE NUMBER OF
                                        ; PRINTABLE DIGITS TO THE LEFT OF ANY TRAILING 0'S
                                        ; (0-12).  EXP CONTAINS THE BINARY EXPONENT
                                        ; (E1-E3 ARE NO LONGER VALID); EXS HAS THE PROPER
                                        ; SIGN OF THE EXPONENT. SPECIFICALLY, RNDATN
                                        ; DOES THE FOLLOWING:
                                        ;   1) ROUNDS THE NUMBER AT THE POSITION POINTED TO BY A.
                                        ;   2) CONVERTS THE STRING FROM BCD TO ASCII.
                                        ;   3) ENSURES THE SIGNS ARE PROPER WITH RESPECT TO
                                        ;      THE NUMBER VALUE AND THE + MODIFIER.
                                        ;   4) CALCULATES THE NUMBER OF DIGITS TO THE LEFT
                                        ;      OF ANY TRAILING ZEROS AND STORES THE RESULT IN DIGCNT.

07A0 97 00G RNDATN: STA A K6,D          ; SAVE POSITION.
07A2 F6 0000G       LDA B N12           ; GET ZERO INDICATOR.
07A5 26 0A          BNE  B54            ; NOT ZERO, DO ROUNDING.
07A7 BD 08BE' RTX:  JSR  ROUND          ; SET UP ZERO NUMBER.
07AA B7 0000G       STA A EXP           ; CLEAR BINARY EXPONENT.
07AD B7 0000G       STA A EXS
07B0 39             RTS                 ; RETURN.


07B1 BD 07FB' B54:  JSR  CONEXP         ; CONVERT & CHECK EXPONENT
07B4 24 03          BCC  B54A           ; OK EXPONENT.
07B6 7E 0819'       JMP  PLER14         ; EXPONENT TOO LARGE.
07B9 96 00G B54A:   LDA A EXH,D         ; GET EXPONENT.
07BB D6 00G         LDA B EXL,D
07BD CB 01          ADD B 1,I           ; MAKE ABSOLUTE.
07BF 89 00          ADC A 0,I
07C1 D8 00G         ADD B K6,D          ; CALCULATE ROUND POSITION.
07C3 89 00          ADC A 0,I
07C5 2B 1E          BMI  TRUE0          ; DIGITS TOO FAR RIGHT.
07C7 26 26          BNE  RND12          ; DIGITS TOO FAR LEFT.
07C9 5D             TST B               ; SET Z FOR B.
07CA 26 1E          BNE  CK12           ; CHECK FOR TOO FAR LEFT.
07CC 96 00G         LDA A X5,D          ; 0=CHECK FOR 5 OR OVER TO  IMMEDIATEL RIGHT.
07CE 2A 15          BPL  TRUE0          ; LESS THAN 5, GIVE TRUE 0.
07D0 86 01          LDA A 1,I           ; GET DIGIT COUNT OF 1.
07D2 97 0000G       STA A DIGCNT        ; SET UP FOR PRINT.
07D4 86 31          LDA A '1,I          ; GET ASCII 1.
07D6 B7 0000G       STA A N12           ; SET UP FOR PRINT.
07D9 DE 00G         LDX  EXH,D          ; BUMP UP EXPONENT.
07DB 08             INX
07DC DF 00G         STX  EXH,D
07DE 8D 1B          BSR  CONEXP         ; RESET EXP,EXS.
07E0 BD 04D2'       JSR  RSN            ; SET UP PFLG/SIGN.
07E3 39             RTS                 ; RETURN.
```

```
    1  07E4  7F  0000G     TRUE0:  CLR   N12     ; CLEAR 0 INDICATOR.
    2  07E7  20  8D                BRA   RTX     ; LET ROUND FINISH TRUE 0.
    3  07E9  86  0C        CK12:   LDA A 12.,I   ; GET MAX DIGIT LIMIT.
    4  07EB  10                    SBA           ; COMPARE TO REQUESTED.
    5  07EC  24  04                BCC   2NDAT   ; 12 OR LESS, RND AT INDICATED PLACE.
    6  07EE  86  0C        RND12:  LDA A 12.,I   ; GET MAXIMUM DIGIT COUNT.
    7  07F0  20  01                BRA   RNDT    ; GENERATE MAXIMUM DIGITS.
    8  07F2  17            2NDAT:  TBA           ; A = # OF DIGITS.
    9  07F3  8D  048F'      RNDT:   JSR   ROUND   ; ROUND AT PLACE POINTED TO BY 'A'.
   10  07F6  8D  03                JSR   CONEXP  ; UPDATE EXP,EXS, & CK FOR OVERFLOW.
   11  07F8  25  1F                BCS   PLER14  ; OVERFLOW.
   12  07FA  39                    RTS
```

```
                                        ; CONEXP CONVERTS THE 16 BIT TWO'S COMPLIMENT
                                        ; EXPONENT TO AN ABSOLUTE VALUE AND SIGN IN EXP AND EXS
                                        ; RESPECTIVELY.  IF THE ABSOLUTE VALUE OF THE EXPONENT
                                        ; IS GREATER THAN 127, AN ERROR EXIT IS MADE.

07E8  7F  0000G     CONEXP: CLR   EXS
07FE  D6  00G               LDA B EXL,D       ; GET EXPONENT.
0800  96  00G               LDA A EXH,D
0802  2A  09                BPL   CONP        ; POSITIVE, DO NOT NEGATE.
0804  87  0000G             STA A EXS         ; SET SIGN =-.
0807  43                    COM A             ; NEGATE EXPONENT.
0808  53                    COM B
0809  CB  01                ADD B 1,I
080B  89  00                ADC A 0,I
080D  26  08        CONP:   BNE   CNPERR      ; CHECK HIGH ORDER.
080F  50                    TST B             ; SET UP LOW ORDER CHECK.
0810  2B  05                BMI   CNPERR      ; TWO LARGE.
0812  F7  0000G             STA B EXP         ; SAVE ABSOLUTE VALUE.
0815  0C                    CLC               ; CLEAR ERROR FLAG.
0816  39                    RTS               ; RETURN.
0817  00          CNPERR: SEC                 ; SET ERROR FLAG.
0818  39                    RTS               ; ERROR RETURN.
0819  32          PLER14: PUL A             ; SCRATCH RETURN.
081A  32                    PUL A
081B  86  00G       LER14:  LDA A ERDEOR,I    ; GET ERROR CODE.
081D  7E  008C'             JMP   EREXIT      ; TAKE ERROR EXIT.


                                        ; CALCOM IS USED TO CALCULATE THE NUMBER OF
                                        ; COMMAS NEEDED TO BREAK UP THE LEFT DIGITS FIELD.

                                        ; IF COMFLG IS NOT SET OR IF ES IS NEGATIVE (-),
                                        ; AN IMMEDIATE EXIT IS MADE LEAVING COMCNT&LCOM EQUAL TO 0.

                                        ; ON EXIT, COMCNT WILL CONTAIN THE TOTAL NUMBER
                                        ; OF COMMAS REQUIRED AND LCOM WILL CONTAIN THE
                                        ; REMAINDER FACTOR (LEFT-MOST DIGIT COUNT BEFORE
                                        ; THE FIRST COMMAS IS OUTPUT).


0820  96  00G       CALCOM: LDA A COMFLG,D    ; GET COMMA FLAG.
0822  26  01                BNE   CLP         ; SET, CHECK SIGN.
0824  39                    RTS               ; NOT SET, RETURN.
0825  B6  0000G     CLP:    LDA A EXS         ; GET SIGN.
0828  2A  01                BPL   CLOP        ; PLUS, DO CALCULATION.
082A  39                    RTS               ; NO LEFT DIGITS, EXIT.
082B  F6  0000G     CLOP:   LDA B EXP         ; GET EXPONENT (LOOP COUNT).
082E  4F                    CLR A             ; CLEAR COMMA COUNTER.
082F  5C                    INC B             ; MAKE COUNT ABSOLUTE.
0830  5A            CLP2:   DEC B             ; SUBTRACT 3 FROM COUNT.
0831  5A                    DEC B
0832  5A                    DEC B
0833  27  05                BEQ   CCFIN       ; TEST FOR LOOP OVER.
0835  2B  03                BMI   CCFIN
```

```
 58  0837  4C                    INC A                   ; BUMP UP COMMA COUNT.
 59  0838  20  F6                BRA     CLP2            ; CONTINUE LOOP.
 60  083A  40              CCFIN: TST A                   ; SET Z.
 61  083B  26  01                BNE     CC2             ; NOT 0, RESTORE REMAINDER.
 62  083D  39                    RTS                     ; NO COMMAS. RETURN.
 63  083E  5C              CC2:  INC B                   ; RESTORE REMAINDER.
 64  083F  5C                    INC B
 65  0840  5C                    INC B
 66  0841  F7  0000G             STA B   LCOM            ; SAVE REMAINDER
 67  0844  B7  0000G             STA A   COMCNT          ; SAVE # OF COMMAS
 68  0847  39                    RTS                     ; RETURN
 69        0001'                 .END
```

SYMBOL TABLE

AFIN	03A0R	AFO	0381R	AGET	0384R	ALOOP	0396R	AOUT	0390R
APNT	0056R	ASR	0200R	AQ	058ER	BMOD	030ER	BX1	0628R
B0	007ER	B1	018DR	B111	01A4R	B13	00B0R	B140	04F2R
B141	04ECR	B142	04F6R	B17	01F8R	B18	01FFR	B19	0253R
B2	00CCR	B22	0283R	B24	02E0R	B25	02E4R	B25N	0249R
B27	0357R	B27A	0354R	B28	036CR	B28A	0361R	B29	037DR
B3	0004R	B30	0385R	B31	038AR	B32	038AR	B33	03DAR
B34	03D0R	B35	03E7R	B36	03E29	B4	017CR	B41	0696R
B42	0687R	B43	06CAR	B44	06FER	B44C	0700R	B5	017DR
B54	07B0R	B54A	07B8R	B6	01ABR	B60	05C7R	B61	05D8R
B62	05DER	B63	0612R	B64	0634R	B65	063CR	B66	0667R
B67	0654R	B7	01EBR	B8	0084R	B9	0088R	CALCOM	0820R
CCFIN	083AR	CC2	083ER	CHKF	0568R	CHKF2	057CR	CKDP	0620R
CKFORN	05E4R	CKNUM	0172R	CK12	07E9R	CLNER	00AER	CLOP	0828R
CLP	0825R	CLP2	0830R	CNPERR	0817R	COMCNT=	xxxxxx G	COMFLG=	xxxxxx G
COMMA	0233R	COMOUT	06D2R	CONEXP	07FBR	CONP	080DR	COUTDP	071ER
CR	0268R	CS	034ER	CURSER=	xxxxxx G	DDP	= xxxxxx G	DEFALT	0521R
DEFCK	0515R	DEFFIN	0021R	DEFPNT	0008RG	DEND	058ER	DEXIT	074BR
DFM	0046R	DFNUM	0028R	DI	04E3R	DIGCNT=	xxxxxx G	DIGOUT	0691R
DL	= xxxxxx G	DN1	= xxxxxx G	DN2	= xxxxxx G	DOLFLG=	xxxxxx G	DOLLAR	0227R
DOLOUT	06A0R	DON2	05A0R	DP	= xxxxxx G	DPCONT=	xxxxxx G	DPOUT	071AR
DT	= xxxxxx G	DZN	0735R	ECONT	045ER	ECONT1	0487R	ECONT2	0484R
ECONT3	047EP	EMD	0617R	EOLP	044DR	EOUT	0425R	EOUTS	0439R
ERBCMU=	xxxxxx G	ERBDPU=	xxxxxx G	ERBMDU=	xxxxxx G	ERBPNU=	xxxxxx G	ERDATM=	xxxxxx G
ERDECR=	xxxxxx G	EREXIT	008CR	ERF	02D7R	ERFLDO=	xxxxxx G	ERIFC =	xxxxxx G
ERINFS=	xxxxxx G	ERINU =	xxxxxx G	ERLRSU=	xxxxxx G	ERNEST=	xxxxxx G	EROUOT=	xxxxxx G
ERRCD =	xxxxxx G	ERTABO=	xxxxxx G	ERTMOS=	xxxxxx G	ES	= xxxxxx G	ETMSD	043CR
EXH	= xxxxxx G	EXIT	008ER	EXL	= xxxxxx G	EXP	= xxxxxx G	EXS	= xxxxxx G
E1	= xxxxxx G	E2	= xxxxxx G	E3	= xxxxxx G	FFLG	= xxxxxx G	FINCK	0606R
FLAGCK	01EER	FLAGS	= xxxxxx G	FLAG8	01C3R	FLG1	01E7R	FMTCLN	0090RG
FMTFLG=	xxxxxx G	FMTINI	005FRG	FMTOUT	007FR	FMTPNT	0000RG	FN	0095R
F1	= xxxxxx G	F2	= xxxxxx G	GETCH	0185R	GETCHI	0186R	GETN	00DDR
GETNUM	0123R	INCCUR	01AER	ISL	= xxxxxx G	ISP	= xxxxxx G	JP	0209R
JPER	0224R	K0	= xxxxxx G	K1	= xxxxxx G	K2	= xxxxxx G	K3	= xxxxxx G
K4	= xxxxxx G	K5	= xxxxxx G	K6	= xxxxxx G	LCOM	= xxxxxx G	LDIG	= xxxxxx G
LDIGA =	xxxxxx G	LDIGCK	06AFR	LER14	0818R	LF	0264R	LI	0311R
LICONT	0348R	LILOOP	031CR	LIOUT	032AR	LITCK	032FR	LITFIN	0330R
LITFLG=	xxxxxx G	LITRY	0325R	LOOP	024FR	LRSERR	0293R	LRSFLG=	xxxxxx G
LSPS	= xxxxxx G	L1	= xxxxxx G	L2	= xxxxxx G	L3	= xxxxxx G	L4	= xxxxxx G
MDERR	0153R	MONER	0159R	MFLG	= xxxxxx G	MINUS	0217R	MODERR	023ER
MULER	017JR	MUL10	015ER	NE	0404R	NECONT	041DR	NEXIT	014ER
NMCK	0411R	NOP	0242R	NP	0260R	NPNT	005AR	NRMOUT	003CR
NS	= xxxxxx G	NUMERR	0157R	NUMFLG=	xxxxxx G	NZ	058FR	N1	= xxxxxx G
N11	= xxxxxx G	N12	= xxxxxx G	OUTIT	0728R	PARCNT=	xxxxxx G	PER	02D2R
PERIOD	0222R	PERJ	02CFR	PEX	02D4R	PFLG	= xxxxxx G	PLER14	0819R
PLUS	020CR	PRINTF=	0000RG	PTR	01C1R	PUTBYT=	xxxxxx G	PUTCHR	01B4R
RDIG	= xxxxxx G	RDIGA	= xxxxxx G	RECURS	00E0R	RECUR1	0085R	RECUR2	009CR
RNDAT	07F2R	RNDATA=	xxxxxx G	RNDATN	079FR	RNDLP	04A2R	RNDN	04B5R
RNDRET	04B4R	RNDT	07F3R	RNDO	04AAR	RND12	07EER	ROUND	048FR
RS	0297R	RSN	04D2R	RSNLP	04C7R	RSNO	04CFR	RSNR	04E2R
RTRNTG=	xxxxxx G	RTX	07A6R	SCR	0288R	SEARCH	008FR	SETDF	0773R
SETF2	0586R	SETOK	079ER	SETZ	0581R	SF	03D4R	SFOUT	004CR
SFPNT	0058R	SGNOUT	06B7R	SLOOP	00C4R	SLP	03A1R	SP	024DR
SPACES	039FR	SPR	0250R	SP1	= xxxxxx G	SP2	= xxxxxx G	SP3	= xxxxxx G
SP4	= xxxxxx G	SYNTX1	0526R	SYNTX2	0549R	TAB	026CR	TABCNT=	xxxxxx G
TABERR	0284R	TABLE	00E7R	TABON	0280R	TF2	05A6R	TRUEO	07E4R
UPCASE=	xxxxxx G	X0	= xxxxxx G	Y1	= xxxxxx G	X2	= xxxxxx G	X3	= xxxxxx G

```
PRINTF    PRINT STATEMENT FORMA        RT-11 MMAC  VM02  0   14-OCT-76 01:38:17 PAGE 32+
SYMBOL TABLE

X4    = XXXXXX G      X5    = XXXXXX G      ZERCNT= XXXXXX G      ZERLP    0764R      ZEROUT   0756R
ZERO1 = XXXXXX G      ZERO2 = XXXXXX G      ZERO3 = XXXXXX G      ZERSWH= XXXXXX G      ZERSWL= XXXXXX G
Z0        0759R       ZRCK      06F4R
. ABS.    0000      00
          0848      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2308.  WORDS
.SY:PRINTF/C(DK1:SEICL1,PRINTF
```

CROSS REFERENCE TABLE (CREF V01-03)

	1-18			
AD	26-22	26-26#		
AFIN	22-37	22-57	22-64#	
AEO	22-42#	22-48		
AGET	22-52	22-71#		
ALOOP	22-52#	22-55		
AOUT	22-39	22-50#		
APNT	5-18	5-47#		
ASR	9-67	15-12#		
B0	6-20	6-23#		
B1	12-17	12-20#		
B111	12-29	12-31#		
B13	8-24	8-27#		
B140	25-13#	25-18		
B141	25-11	25-14#		
B142	25-16	25-19#		
B17	14-42	14-46#		
B18	14-47	14-50#		
B19	17-12#	17-15	17-47	
B2	9-17	9-20#		
B22	19-12	19-15#		
B24	20-19	20-23#		
B25	20-24	20-26#		
B25N	16-10#			
B27	22-19	22-22#		
B27A	22-21#	22-26	22-31	22-43
B28	22-29	22-32#		
B28A	22-24	22-27#		
B29	22-35	22-40#		
B3	9-21	9-26#		
B30	22-41	22-44#		
B31	22-45	22-47#		
B32	22-72	22-75#		
B33	23-14#	23-19	23-24	23-46
B34	23-12	23-15#		
B35	23-17	23-20#		
B36	23-22	23-25#		
B4	11-16#	11-18		
B41	28-5#	28-8		
B42	28-18#	28-25		
B43	28-21	28-26#		
B44	28-49#	28-54	28-60	
B44C	28-48	28-56#		
B5	11-14	11-17#		
B54	30-19	30-26#		
B54A	30-27	30-29#		
B6	12-35	12-37#		
B60	26-33#			
B61	26-35	26-43#		
B62	26-41	26-45#		
B63	26-67	26-70#		
B64	27-10	27-12#	27-22	
B65	27-8	27-16#		
B66	27-17	27-39#		
B67	27-28#	27-30		
B7	14-27	14-30#		

CROSS REFERENCE TABLE (CREF V01-03)

```
B8      8-22    8-29#
B9      8-31#
BMOD    20-15   20-41#
BX1     27-5    27-7#
CALCOM  26-16   32-44#
CC2     32-61   32-63#
CCFIN   32-56   32-57   32-60#
CHKF    25-64   25-66#
CHKF2   25-70   25-73#
CK12    30-38   31-3#
CKDP    26-49   26-71   27-4#
CKFC    26-24   26-48#
CKN     9-26    10-19   10-25   10-31   11-13#  25-53   25-56
CLMEK   8-26#   8-28
CLOP    32-48   32-50#
CLP     32-45   32-47#
CLP2    32-53#  32-59
CNPERR  32-17   32-19   32-23#
COMCNT  3-34#   26-54   29-12%  32-67%
COMFLG  1-76#   6-16%   14-22   15-43%  28-88%  32-44
COMMA   9-77    15-40#
COMOUT  28-16   28-30#  28-38   28-43
CONEXP  26-8    27-4    30-26   30-48   31-10   32-8#
CONP    32-11   32-17#
COUTDP  28-65#
CR      9-53    17-27#
CS      9-43    22-18#
CURSER  1-59#   6-17%   6-18%   12-15   12-20   12-21   12-40   12-42%  19-25   20-21%  21-11   21-34%
DDP     3-8#    25-20%  25-42%  28-62
DEFALT  25-32   25-36#
DEFCK   25-27   25-31#
DEFFIN  5-21#   5-37    5-44
DEFPNT  1-45#   5-11#
DEND    26-39   26-41   26-45   26-52   26-60   26-65   26-67   26-77   26-4#
DEXIT   28-63   28-68   28-73   28-84#
DFM     5-30    5-38#
DFNUM   5-17    5-26#
DI      9-45    25-10#
DIGCNT  3-11#   24-19%  24-35%  24-37   24-46   25-21%  26-18   26-37   26-75   27-7    27-42   27-52   30-42%
DIGOUT  27-11   27-14   27-33   27-36   27-42   27-49   27-51   27-53   28-3#
DL      1-66#   22-32   22-36   22-40   22-47
DN1     3-9#    25-22%  25-28%  25-37%  26-65
DN2     3-10#   25-23%  25-50%  25-63%  26-11   27-12   27-32   27-48
DOLFLG  1-75#   6-15%   14-20   15-37%  28-9    28-87%
DOLLAR  9-75    15-33#
DOLOUT  28-4    28-9#
DON2    26-5    26-11#
DP      1-65#   22-79   22-80
DPCONT  3-16#   5-3%    5-12%   23-58   23-80   28-65
DPOUT   28-28   28-46   28-52   28-62#
DT      1-67#   5-15    22-22   23-15   25-14
DZN     28-74#  28-80
E1      2-12#   23-91   24-22
E2      2-13#   23-84   23-89
E3      2-14#   23-79
ECONT   23-71   23-75#
ECONT1  23-87   23-91#
```

CROSS REFERENCE TABLE (CREF V01-03)

```
ECONT2  23-86   23-90#
ECONT3  23-81   23-83   23-88#
EMD     26-33   26-74#  27-21
EOLP    23-67#  23-74
EOUT    23-32   23-34   23-40   23-52#
EOUTS   23-57   23-60#
ERBCMU  1-30#   16-10
ERBDPU  1-31#   15-30
ERBMDU  1-28#   8-25    10-36   15-46   22-30   23-23   25-71
ERBPNU  1-27#   8-27    19-38
ERDATM  1-33#   22-25   23-18   25-17
ERDEOR  1-35#   32-27
EREXIT  7-18#   8-26    9-29    10-41   15-31   19-39   21-18   22-21   23-14   25-13   25-72   26-69   32-28
ERF     9-61    20-13#
ERFLDO  1-34#   22-42   26-68
ERIFC   1-24#   9-28
ERINFS  1-23#   7-17    19-13
ERINU   1-25#   10-38   23-45
ERLRSU  1-29#   18-11
ERNFST  1-22#   6-21
ERQUOT  1-32#   21-17
ERRCO   1-48#   6-22#   7-18#   8-11    13-10
ERTABO  1-36#   17-48
ERTMOS  1-26#   22-20   23-13   25-12
ES      2-11#   23-27   24-31#
ETMSD   23-59   23-61#
EXH     3-14#   5-29    24-17#  30-29   30-45   30-47#  32-10
EXIT    7-19#   8-30    8-33
EXL     3-15#   5-28    24-18#  30-30   32-9
EXP     3-17#   26-31#  26-43   26-74   27-20#  27-26   27-39   30-21#  32-20#  32-50
EXS     3-18#   26-28   27-16   30-22#  32-8#   32-12#  32-47
F1      3-12#   25-24#  25-33#  26-48
F2      3-13#   25-25#  25-80#  26-4    27-9    27-46
FFLG    1-72#   6-12#   10-14   14-14   15-15#  22-65#  23-25   23-53#  25-31   25-34#  25-76#
FINCK   26-58   26-60   26-62   26-65#
FLAG8   8-20    14-7#   14-40   15-12   15-18   15-24   15-33   15-40   16-7    20-13   22-27   23-20
FLAGCK  14-40#  17-11   17-37   19-10   21-10
FLAGS   1-80#   14-24#
FLG1    14-9    14-11   14-13   14-15   14-17   14-19   14-21   14-23   14-27#
FMTCLN  1-42#   8-11#
FMTFLG  1-77#   7-14#   8-17#   9-31    22-18   23-11   25-10
FMTINI  1-40#   5-22    6-7#
FMTOUT  5-6     5-23    7-11#
FMTPNT  1-41#   5-3#
FN      8-12    8-14#
GETCH   10-17   10-23   10-29   12-15#  12-34   18-6    19-11   21-23   25-38   25-44   25-58   25-69
GETCH1  7-15    8-18    12-34#  21-15   21-27   25-55
GETN    9-27    9-30#
GETNUM  9-30    10-12#  25-57
INCCUR  10-22   10-28   12-37   12-40#  25-43   25-51   25-61   25-68   25-78
ISL     1-64#   5-21#   6-19    12-16
ISP     1-63#   5-19#   5-35#   5-42#   12-22   12-23
JP      15-16#  15-22   15-28   15-38   15-44   16-9
JPER    15-31#  15-47   16-11
K0      2-2#    12-24#  12-26   17-10#  17-12   17-39#  22-50#  22-77   22-86   22-88#  23-67#  23-72   26-39   26-56#
        26-63#  26-66   26-76#  27-25#  27-34
K1      2-3#    12-25#  22-51#  22-78
```

CROSS REFERENCE TABLE (CREF V01-03)

K2 2-4# 22-33% 22-21 22-76%
K3 2-5#
K4 2-6# 21-12% 21-33 22-38% 22-49% 22-56 22-60%
K5 2-7# 22-81% 22-83
K6 2-8# 22-82% 23-48% 23-51 23-54% 23-65% 23-70% 30-17% 30-33
L1 1-50# 19-31 19-34% 20-18% 20-33%
L2 1-52# 19-29 19-32% 20-32 20-35%
L3 1-54# 19-27 19-30% 20-34 20-37%
L4 1-56# 19-28% 20-36
LCOM 3-23# 28-15 28-37% 28-42% 28-47% 28-53% 28-59% 29-11% 32-66%
LDIG 3-21# 26-21% 26-38% 26-44% 26-51 28-20% 28-35% 29-10%
LDIGA 3-22# 26-23% 26-46% 28-17 28-22 28-24% 28-30 28-33%
LDIGCK 28-13 28-15#
LER14 26-10 27-6 32-27#
LF 9-55 17-23#
LL 9-49 21-10#
LICONT 21-30 21-33#
LILOOP 21-15# 21-22 21-35
LIOUT 21-21# 21-28
LITCK 21-20 21-23#
LITFIN 21-24 21-26 21-29#
LITFLG 1-69# 6-9% 12-28 21-14% 21-31%
LITRY 21-16 21-19#
LOOP 17-10# 17-20 17-24 17-28
LRSERR 18-7 18-11#
LRSFLG 1-70# 6-10% 8-29 14-8 18-9%
LSPS 3-20# 26-70% 28-3 28-7% 29-9%
MDERR 10-15 10-36#
MDNER 10-37 10-39#
MFLG 1-74# 6-14% 14-18 15-27% 26-57 28-86% 29-19
MINUS 9-71 15-24#
MODERR 14-45 15-14 15-20 15-26 15-35 15-42 15-46# 20-41
MUL10 10-21 10-27 10-43#
MULER 10-45 10-47 10-49 10-51 10-53 11-1#
N1 2-15# 24-39
N11 2-16# 23-66
N12 2-17# 5-26 23-61 24-15 26-45 27-27 27-44 30-18 30-44% 31-1%
NE 23-26 23-35#
NECONT 23-44 23-47#
NEXIT 10-20 10-26 10-33#
NMCK 23-36 23-41#
NOP 9-65 16-7#
NP 9-63 17-19#
NPNT 5-34 5-55#
NRMOUT 5-27 5-34# 5-40
NS 2-18# 23-55 24-29% 24-49 24-53% 26-61 28-12 29-21 29-24%
NUMERR 10-13 10-18 10-24 10-30 10-32 10-34 10-38# 11-3
NUMFLG 1-71# 5-11% 10-12 10-16% 10-33 10-43 10-48 10-54% 14-12 14-46 14-49% 17-14% 17-40 17-44%
17-46# 19-33 19-35% 21-29% 22-44 22-66% 23-35 23-41% 25-26 25-29% 25-62 25-64%
NZ 26-19 26-28#
OUTIT 28-66 28-69#
PERCNT 1-60# 6-8% 14-10 19-16 19-18% 20-23%
PER 19-17 19-38# 20-17
PERIOD 9-73 15-30#
PERJ 19-36# 20-22 20-25 20-38
PEX 17-49 18-12 19-14 19-39#
PFLG 1-73# 6-13% 14-16 15-21% 23-52% 24-47 26-59 28-85% 29-17

CROSS REFERENCE TABLE (CREF V01-03)

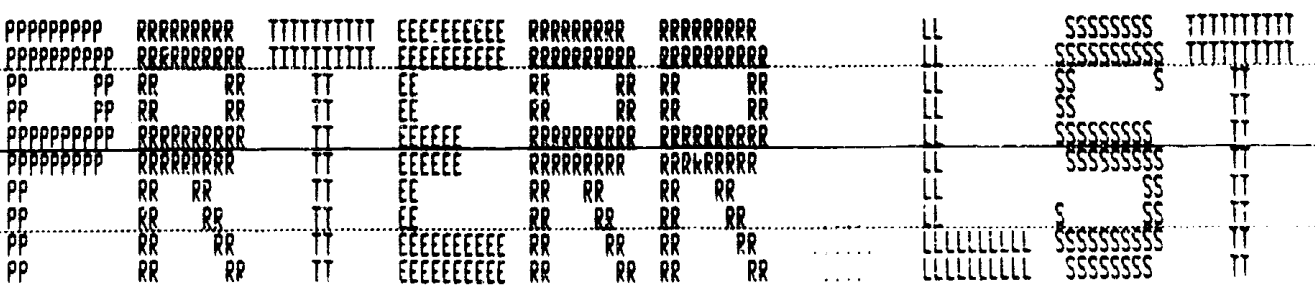
```
PLER14  30-28   31-11   32-25#
PLUS    9-69    15-18#
PRINTF  1-17#   1-18#
PTR     13-11   13-15#
PUTBYT  1-43#   13-9
PUTCHR  8-32    13-7#   17-13   21-21   22-54   22-59   23-60   23-62   23-64   23-69   23-76   23-78   23-88   23-90
        23-92   28-6    28-11   28-14   28-19   28-34   28-40   28-50   28-57   28-69   28-78   28-100
RDIG    3-26#   27-24#  27-43#  28-67   28-72   28-79#  29-13#
RDIGA   3-27#   27-31#  27-45#  28-74   28-77#
RECUR1  7-15#   9-33
RECUR2  8-18#   9-72
RECURS  9-31#   9-79    15-16   17-16   19-36   21-32
RNDO    24-25   24-28#
RNDI2   30-36   31-6#
RNDAT   31-5    31-8#
RNDATA  1-44#   24-34
RNDATN  26-12   30-17#
RNDLP   24-23#  24-27
RNDN    24-16   24-34#
RNDRET  24-32#
RNDT    31-7    31-9#
ROUND   23-28   23-38   23-50   24-15#  26-7    30-20   31-9
RS      9-59    19-10#
RSN     24-38   24-47#  30-49
RSNLP   24-41#  24-45
RSNO    24-43   24-46#
RSNR    24-48   24-51   24-54#
RTRNTG  1-38#   5-4     5-13    7-11    8-14    13-7
RTX     30-20#  31-2
SCR     9-57    18-6#
SEARCH  7-16    8-19    9-14#
SETDF   26-15   29-8#
SETF2   25-74   25-78#
SETOK   29-18   29-20   29-23   29-25#
SETZ    26-20#  26-29
SF      9-47    23-11#
SFOUT   5-33    5-41#
SFPNT   5-41    5-51#
SGNOUT  28-10   28-12#
SLOOP   9-16#   9-25
SLP     22-57#  22-61
SP      9-51    17-9#
SP1     1-51#   19-23   19-26#  20-20   20-27#
SP2     1-53#   19-21   19-24#  20-26   20-29#
SP3     1-55#   19-19   19-22#  20-28   20-31#
SP4     1-57#   19-20#  20-30
SPACES  22-53   22-56#
SPR     17-16#  17-45   18-10
SYNTX1  25-30   25-35   25-38#
SYNTX2  25-48   25-53#
TAB     9-81    17-37#
TABCNT  1-61#   17-41
TABERR  17-42   17-48#
TABLE   9-14    9-42#
TABON   17-43   17-46#
TF2     26-9    26-15#
TRUEC   30-35   30-40   31-1#
```

CROSS REFERENCE TABLE (CREF V01-03)

UPCASE	1-46#	12-30					
X0	3-1#						
X1	3-2#						
X2	3-3#						
X3	3-4#						
X4	3-5#						
X5	3-6#	30-39					
ZERCNT	3-30#	28-94#	28-101#				
ZERLP	28-99#	28-102					
ZERO1	3-24#	26-40#	26-52	28-26	28-46	28-61#	29-14#
ZERO2	3-25#	27-13#	27-41#	27-50	28-70	29-15#	
ZERO3	3-23#	27-35#	27-54#	28-81	29-16#		
ZEROUT	28-27	28-71	28-82	28-92#			
ZERSUH	3-31#	28-96#	28-103				
ZERSUL	3-32#	28-98#					
ZO	28-92	28-94#					
ZRCK	28-36	28-45#					

CROSS REFERENCE TABLE (CREF V01-03)

SEL 1-3#



PRTERR ERROR CODE LISTING WITH RT-11 MMAC VM02-10 14-OCT-76 01:38:57
TABLE OF CONTENTS

```
                                .TITLE  PRTERR  ERROR CODE LISTING WITH COMMENTS
                                .IDENT  /JGA138/
                        ;
                        ;       THIS IS THE SYSTEM ERROR MESSAGE WRITER.  IT GETS IT'S INPUT
                        ;       FROM ERRCD, ERRCDB, ERCTRA, ERCTRB AND THE OBJECT LINES.
                        ;       IT CAN BE CALLED FROM A ROM PACK DIRECTLY BUT IF SO
                        ;       THE BASIC PROGRAM THAT WAS RUNNING WILL NOT BE STOPED.
                        ;       THERE ARE TWO SPECIAL ERROR CODES FOR ERROR REPORTING FROM
                        ;       ROM PACKS.  THESE ARE ERROM1 AND ERROM2.
                        ;       IF A ROM PACK WANTS TO STOP EXECUTION IT MUST SET THE ERROR
                        ;       CODE IN ERRCD AND RETURN TO IT'S CALLER.
                        ;
0000                            .ASECT                  ;PUT IT BELOW PIAS FOR NOW
8358                            . = ^O8358
                        ;
                                .SBTTL  XXX MACRO DEFINITIONS
                        ;
                        ;       DEFINE AN ERROR NAME AND MESSAGE CODE MACRO
                        ;
                                .MACRO  ERR     NAME,NUM
                        NAME    =       ERRCTR
                                .BYTE   NUM
                                .GLOBL  NAME
                        ERRCTR  =       ERRCTR+1
                                .ENDM
                        ;
                        ;       DEFINDE DUPLICATE ERROR CODE
                        ;
                                .MACRO  DERR    NAME,NUM
                        NAME    =       ERRCTR-1
                                .GLOBL  NAME
                                .ENDM
                        ;
                        ;       GENERATE A MESSAGE NAME, WORD CODES AND COMMANDS
                        ;
                                .MACRO  MSG     ID,W1,W2,W3,W4,W5,W6,W7,W8
                        M'ID    =       .-MSGORG
                                .BYTE   W1
                                .IF     NB,W2
                                .BYTE   W2
                                .IF     NB,W3
                                .BYTE   W3
                                .IF     NB,W4
                                .BYTE   W4
                                .IF     NB,W5
                                .BYTE   W5
                                .IF     NB,W6
                                .BYTE   W6
                                .IF     NB,W7
                                .BYTE   W7
                                .IF     NB,W8
                                .BYTE   W8
                                .ENDC
                                .ENDC
                                .ENDC
                                .ENDC
                                .ENDC
```

```
XXX MACRO DEFINITIONS

                        .ENDC
                        .ENDC
                        .ENDM
                ;
                ;       GENERATE     PRESSED WORD TABLES
                ;
                        .MACRO  .WRD    NAME,TEXT
NAME            =       (.-WRDORG)/2
CTR             =       3
PACKED          =       1
                        .IRPC   CHR,TEXT
PACKED          =       PACKED*32.+L'CHR
CTR             =       CTR-1
                        .IF     EQ,CTR
CTR             =       3
                        .WORD   PACKED
PACKED          =       0
                        .ENDC
                        .ENDM
                        .IF     NE,CTR-3
PACKED          =       PACKED*32.
                        .IF     NE,CTR-1
PACKED          =       PACKED*32.
                        .ENDC
                        .WORD   PACKED
                        .ENDC
                        .ENDM
                ;
                ;       DEFINE A COMMAND
                ;
                        .MACRO  CMD     NAME
C'NAME          =        -CMDTBL+240
                        .BYTE   X'NAME-CMDORG
                        .ENDM
```

```
                                .SBTTL  XXX DEFINE THE ERROR NAMES AND ASSIGN CODE NUMBERS
                                ENABL   LC
                                .GLOBL  SIZERR
                        0001    ERRCTR  =       1
8358                            MSGTBL: ERR     EREPOV,M1
                                                ;FPMATH FLOATING POINT OVERFLOW,
                                                        ;RETURNS + OR - MACHINE INFINITY
8359                            ERR     EREPDV,M1
                                                ;FPMATH FLOATING PT DIVIDE BY ZERO WAS TRIED
835A                            ERR     ERUP,M1
                                                ;FPMATH A.B ERROR, RESULT OUT OF RANGE, OR
                                                        ;A=B=0 (SIZE ERROR IS GENERATED)
8358                            ERR     EREXP,M1
                                                ;FPMATH E.X CAUSED RESULT TO OVER/UNDERFLOW,
                                                        ;I.E., !X!)709, RETURNS +00 OR 0
835C                            ERR     ERTRNG,M1
                                                ;ADMATH TRIG RANGE EXCEEDED, RETURNS 0
835D                            ERR     ERSQR,M1
                                                ;ADMATH NEGATIVE SQUARE ROOT ATTEMPTED
                                                        ;RETURNS +SQR
                        0006    SIZERR  =       ERRCTR-1
835E                            ERR     ERLNNO,M6
                                                ;TRANSL LINE NUMBER OUT OF RANGE, I.E.,
                                                        ; <=0 OR )65,535
835F                            ERR     ERSHAP,M7
                                                ;MATOPR ARRAYS ARE NOT CONFORMABLE
8360                            ERR     ERDIM,M8
                                                ;DIMSUB NO VALUE THERE,
                                                        ;VALUE OUT OF RANGE,
                                                        ;CAN'T DIMENSION A SIMPLE VARIABLE
                                                        ;WITHOUT DELETING IT FIRST.
8361                            ERR     ERSUBC,M8
                                                ;DIMSUB NO VALUE THERE,
                                                        ;VALUE OUT OF RANGE,
                                                        ;CAN'T SUBSCRIPT A SIMPLE VARIABLE,
                                                        ;ARRAY/SUBSCRIPT MISMATCH
8362                            ERR     ERNOFN,M9
                                                ;EVLEN  FNX NOT DEFINED
8363                            ERR     ERCLAI,M10
                                                ;ROM-PACK CALL ARGUMENT ERROR
8364                            ERR     ERWBYT,M10
                                                ;BYTE   WRITE BYTE ARGUMENT OUT OF RANGE
8365                            ERR     ERAPP1,M10
                                                ;APPEND PARAMETER OF WRONG TYPE OF OUT OF RANGE
8366                            ERR     ERAPP2,M10
                                                ;APPEND ATTEMPTED TO APPEND AT A NONEXISTANT LINE.
8367                            ERR     ERFUZZ,M10
                                                ;ADMATH INVALID ARGUMENT
8368                            ERR     ERREN,M10
                                                ;RENUMB VALUE OUT OF RANGE,
                                                        ;INTERLACING WILL OCCUR IF RENUMBER
                                                        ;IS ATTEMPTED,
                                                        ;NOT A VALID VALUE
                                                        ;NOTE - CAN OCCUR DURING AN APPEND STMT.
8369                            ERR     ERMARK,M10
836A                            ERR     ERNXTA,M10
                                                ;FLCTRL NEXT STATEMENT IS INVALID
```

XXX DEFINE THE ERROR NAMES AND ASSIGN CODE NUMBERS

```
                                                        ;E.G., NEXT I, WHERE I      RIX, OR
                                                        ;THERE WAS NO FOR TO G      NEXT
8368                    DERR    ERFORA,M10
                                                ;FLCTRL FOR STATEMENT ARGUMENT
                                                        ;E.G., FOR A=3   WHERE A
8366                    DERR    EROFA,M10
                                                ;FLCTRL EXPRESSION TO THE LEFT OF T   IN GOTO...OF.
                                                        (
                                                        ;IS INVALID, E.G., AN ARRAY
8368                    ERR     EREOFN,M10
                                                ;BSTMT  EOF OF UNIT N
836C                    ERR     ERASGN,M11
                                                ;FLDATA ASSIGNMENT ARGUMENT INVALID,
                                                        ;UNDEFINED VARIABLE,
                                                        ;ASSIGNING A MATRIX TO A SCALAR
                                                        ;NONCONFORMABLE ARRAYS,
                                                        ;TRIED TO ASSIGN A STRING THAT WAS
                                                        ;LONGER THAN THE DIMENSIONED RESULT AREA
836D                    ERR     ERUPN,M11
                                                ;ADMATH UP-ARROW ERROR, A B WHERE A<0
                                                        ;AND B WAS NOT AN INTERGER < 256
836E                    ERR     ERLOG,M11
                                                ;ADMATH LOG OR LGT OF A NUMBER <=0WHS
                                                        ;TRIED  RETURNS GARBAGE
836F                    ERR     ERARC,M11
                                                ;ADMATH ASN, ACS & ATN ARG >1
8370                    ERR     ERSET,M11
                                                ;STRING NUMBER OUT OF RANGE, MUST BE
                                                ;  !0(= N <=127
8371                    ERR     ERCARC,M11
8372                    ERR     ERDOMN,M11
                                                ;PGMEVL THE ARGUMENT IS OUTSIDE OF THE
                                                        ;DOMAIN OF THE FUNCTION, I.E.,
                                                        ;STR(A) WHERE A IS AN ARRAY
8373                    ERR     ERREP,M11
8374                    ERR     ERVAL,M11
                                                ;MATOPR NO NUMBER IN THE STRING ARGUMENT.
8375                    ERR     ERMUL,M11
                                                ;MATRIX MATRIX MULTIPLY OPERANDS INVALID
8376                    ERR     ERDET,M11
                                                ;MATRIX INVERSION HAD ZERO DETERMINATE
                                                ;       DATA IN RESULT IS BAD
8377                    ERR     ERCNSF,M12
                                                ;CALL   SPECIFIED CALL NAME NOT FOUND
8378                    ERR     ERNCN,M12
8379                    ERR     ERNDT,M13
                                                ;FLDATA NO DATA STATEMENT OR RESTORE N
                                                        ;WHERE N IS NOT A DATA STATEMENT.
                                                        ;NOT ENOUGH DATA IN DATA STMT.??
837A                    ERR     ERNIMX,M14
                                                ;EVLEN
                                                ;FLCTRL
                                                ;BSTMT  INVALID STATEMENT IN IMMEDIATE MODE,
                                                        ;LINE NUMBER REQUIRED.
837B                    ERR     ERUNDF,M15
                                                ;EVLSUB A VARIABLE WITHOUT AN ASSIGNED
                                                        ;VALUE IS USED IN THE LINE
837C                    ERR     ERNXFN,M16
```

```
                                                        ;PGMEVL AN OPTIONAL EXTENDED FUNCTION
                                                        ;       ROM IS NEEDED TO PEREORM THE FUNCTION
                8370            ERR     ERSEC,M17
                                                        ;       TRIED TO OLD OR APPEND A SECRET PROGRAM
                837E            ERR     ERWSEL,M18
                                                        ;COMPRES MEMORY IS FULL, NO SPACE LEFT FOR
                                                        ;       ALLOCATION.
                                                        ;DIMSUB NOT ENOUGH MEMORY TO DO DIMENSION
                                                        ;PARSE  NOT ENOUGH MEMORY TO TRANSLATE THE LINE
                837F            ERR     ERBRK,M19
                                                        ;EVLEN  BREAK KEY. NEVER PRINTED
                8380            ERR     ERNON1,M20
                                                        ;EVLEN  SIZE ON UNIT.
                8381            ERR     ERNON2,M20
                                                        ;EVLEN  FULL ON UNIT.
                8382            ERR     ERNON3,M21
                                                        ;EVLEN  SRQ ON UNIT.
                8383            ERR     ERNON4,M20
                                                        ;EVLEN  EOI ON UNIT.
                8384            ERR     ERNON5,M22
                                                        ;EVLEN  <EXTERNAL INTERRUPT #1> ON UNIT.
]]]]]
                8385            ERR     ERNON6,M22
                                                        ;EVLEN  <EXTERNAL INTERRUPT #2> ON UNIT.
                8386            ERR     ERNON7,M22
                                                        ;EVLEN  <EXTERNAL INTERRUPT #3> ON UNIT.
                8387            ERR     ERNON8,M23
                                                        ;EVLEN  NO EOFX ON UNIT.
                8388            ERR     ERFBER,M24
                                                        ;LINE TO LONG - UNCOMP
                8389            ERR     ERLOLD,M24
                                                        ;LINE TOO LONG TO INPUT
                838A            ERR     ERLNNF,M25
                                                        ;EVLSUB INDICATED LINE WAS NOT FOUND
                                                        ;USING  USING DOES NOT REFER TO IMAGE STATEMENT
                8388            ERR     ERFNFD,M26
                                                        ;MTCTL  REQUESTED FILE DOESN'T EXIST,
                                                        ;       TRYING TO KILL LAST FILE
                838C            ERR     ERMTRD,M27
                                                        ;MTCTL  READ FAILED AFTER 10 TRIES.
                                                        ;       NOTE - THE READ HEAD IS POSITIONED
                                                        ;       AFTER THE BAD RECORD, SO USER
                                                        ;       CAN CONTINUE TO READ NEXT RECORD
                838D            ERR     EREOM,M28
                                                        ;MTCTL
                                                        ;M8     END OF MEDIUM DETECTED.
                838E            ERR     ERMILA,M28
                                                        ;MTCTL  MAG TAPE FILE ACCESS INVALID.
                838F            ERR     ERWRT,M29
                8390            ERR     ERNCRT,M31
                                                        ;M8
                                                        ;MTCTL  NO MAG TAPE CARTRIDGE PRESENT.
                8391            ERR     ERMEMI,M32
                                                        ;MTCTL  COULDN'T MAKE SENSE OUT OF THE
                                                        ;       MAG TAPE FILE HEADER.
                8392            ERR     ERNOLO,M30
                                                        ;MTCTL  NO PROGRAM TO OLD
                8393            ERR     EROVRD,M32
```

XXX DEFINE THE ERROR NAMES AND ASSIGN CODE NUMBERS

```
                        ;XXXXXX EROVRC IS NOT USED
8394                    ERR     ERNOPN,M33
                                                ;NTCTL  ATTEMPTED TO MARK WHILE A FILE WAS OPEN.
8395                    ERR     ERFILE,M34
8396                    ERR     ERREAD,M34
                                                ;BINCTL BINARY DATA HEADER ERROR
8397                    ERR     ERSTGL M34
                        ;XXXXXX ERSTGL IS NOT USED
8398                    ERR     ERNSE',M34
                        ;XXXXXX ERNSEP IS NOT USED
8399                    ERR     ERATSN,M34
                                                ;IOCTL
                                                ;IECDRV DEVICE SPECIFICATION RANGE ERROR
839A                    ERR     ERNLOD,M34
                                                ;OUTCTL ILLEGAL OPERATION ATTEMPTED ON AN
                                                        ;INTERNAL DEVICE.
839B                    ERR     EREXEC,M34
839C                    ERR     ERIOE,M35
                                                ;IECDRV XXXXXXXXXXXXXXXXX
839D                    ERR     ERQUOT,M36
                                                ;PRINTF INCOMPLETE LITERAL
839E                    ERR     ERNFST,M36
                                                ;PRINTF NO FORMAT STRING
839F                    ERR     ERINFS,M36
                                                ;PRINTF FORMAT STRING IS TOO SHORT, I.E.,
                                                        ;NOT ENOUGH MATCHING DATA SPECIFIERS
83A0                    ERR     ERIFC,M36
                                                ;PRINTF INVALID FORMAT CHARACTER
83A1                    ERR     ERINU,M36
                                                ;PRINTF INVALID NUMBER USAGE, NUMBER OUT OF RANGE
83A2                    ERR     ERTMDS,M36
                                                ;PRINTF TOO MANY DATA SPECIFIERS
83A3                    ERR     ERBPNU,M36
                                                ;PRINTF INVALID PARENTHESIS USAGE
83A4                    ERR     ERBMDU,M36
                                                ;PRINTF INVALID MODIFIER USAGE
83A5                    ERR     ERLRSU,M36
                                                ;PRINTF THE S MUST BE THE LAST CHARACTER
                                                        ;IN THE FORMAT STRING, IF USED.
83A6                    ERR     ERBCMU,M36
                                                ;PRINTF INVALID COMMA USAGE
83A7                    ERR     ERBDPU,M36
                                                ;PRINTF INVALID DECIMAL POINT USAGE
83A8                    ERR     ERDATM,M36
                                                ;PRINTF DATA TYPE MISMATCH.
83A9                    ERR     ERTABO,M36
                                                ;PRINTF TABBING ERROR, ALREADY PAST SPECIFIED POSITION
83AA                    ERR     ERDEOR,M36
                                                ;PRINTF EXPONENT OUT OF RANGE FOR D OPERATOR.
83AB                    ERR     ERISGN,M36
                                                ;USING  IMAGE STRING GONE AFTER PAGE FULL
83AC                    ERR     ERISTS,M36
                                                ;USING  IMAGE STRING TOO SHORT AFTER PAGE FULL
83AD                    ERR     ERDAGN,M36
                                                ;USING  DATA GONE AFTER PAGE FULL
83AE                    ERR     ERFLOO,M37
                                                ;PRINTF FIELD OVERFLOW IN A OR D OPERATOR OUTPUT.
```

```
                ;*** DEFINE THE ERROR NAMES AND ASSIGN CODE NUMBERS

229             83AF                    ERR     ERINTR,M20
230                                             ;       AN INVALED DATA TYPE TRNASFER
231                                             ;WAS TRIED TO AN !NTERNAL DEVICE.
232                                             ;E.G., PRI 2,20: "50,65"
233             8380                    ERR     ERROM1,MROM1
234                                     ;ROM-PACK ANY ERROR IN ROM PACK
235             8381                    ERR     ERROM2,MROM2
236                                     ;ROM-PACK STOP EVAL ONLY ERROR CODE
237             8382                    ERR     ERSTOP,M2
238                                     ;FLCTRL STOP WAS EXECUTED
239             8383                    ERR     ERIERM,M3
240                                     ;EVLEN  THE PROGRAM WAS ABORTED WITH BREAK KEY.
241             8384                    ERR     ERSYNX,M4
242                                     ;PARSE  NOT AN EXPLICIT MESSAGE,
243                                             ;USES THE MAGIC DOWN-ARROW
244             8385                    ERR     ERCHAR,M5
245                                     ;LEX    UNRECOGNIZABLE CHARACTER OR WORD.
246                                             ;NO CLOSING QUOTE
247             8386                    ERR     ERFIXN,M20
248                                     ;FPMATH TRIED TO FIX A NEGATIVE NUMBER.
249                                             ;THE POSITIVE NUMBER IS RETURNED.
250             8387                    ERR     ERFXOV,M20
251                                     ;FPMATH TRIED TO FIX A NUMBER GREATER
252                                             ;THAN 65,535
253                     0060            LASTER  = ERRCTR-1
```

```
                                .SBTTL  XXX SYSTEM VARIABLES & DEFINITIONS
                                ;
                                .GLOBL  ERRCD, ERRCDB, ERCTRA, ERCTRB, HLDEOF
                                .GLOBL  DSPCHR, INAGTE
                                .GLOBL  CLPTR, NLPTR, KEYSTK
                                .GLOBL  TYPIN, EDTPTR, EDTBFR, EDTCLR
                                .GLOBL  FIX1, SCRTCH, TFLGS1, UNLEX
                                .GLOBL  AEX
                                .GLOBL  SRQOFF  ;RESET SRQ PENDING STATE
                                .GLOBL  LDAX, LDBX, LDXX, JMPAX
                                .GLOBL  ANHOLD  ;HOLD AUTO NUMBER - SYNTAX ERROR WITH OPEN BFR
                                .GLOBL  ZX      ;TWO BYTES OF ZEROS IN PAGE ZERO
                                .GLOBL  R5, R9, R10, R11, R12, R13, R16
                                ;
                                ;       CONSTANTS
                                ;
0003                    PGMFP   =       3               ;FOWARD POINTER IN PROGRAM LIST
0007                    PGMLNN  =       7               ;PROGRAM LINE NUMBER DISP
0000                    CR      =       H00             ;CARRIAGE RETURN
0020                    SPACE   =       H20             ;SPACE CHARACTER
007F                    ARROW   =       H7F             ;DOWN ARROW
0083                    RCLKEY  =       H083            ;EDITOR RECALL LINE KEY
                                ;
                                ;       REGISTER USAGE
                                ;
                                ;       R5 - UNLEX INPUT
                                ;       R9 - DITTO
                                ;       R10 - PSUEDO PC. FOR MESSAGE TABLES
                                ;       R11 - WORD LOOP WORK REG
                                ;       R12 - NUMBER TO BE PRINTED BY CNMBR
                                ;       R13 - WORK REGISTER
                                ;
```

```
                                 .SBTTL  *** MAIN ENTRY POINT
                          ;
                                 .GLOBL  PRTERR
                                 .GLOBL  GLBFLG
                          ;
                          ;      PRINTS THE ERROR MESSAGE ASSOCIATED WITH THE ERROR DCODE
                          ;      STORED IN ERRCD.
                          ;
8388   96   00G           PRTERR: LDA A  ERRCD,D        ;LOAD ERROR CODE
838A   27   40                    BEQ    NOERR
838C   96   00G                   LDA A  GLBFLG,D
838E   84   F7                    AND A  -1-8.,I
83C0   97   00G                   STA A  GLBFLG,D       ; RE-ENABLE SRQ INTERUPT
83C2   86   0D                    LDA A  CR,I           ;FORCE CARRIAGE RETURN
83C4   BD   0000G                 JSR    DSPCHR
83C7   7F   0000G                 CLR    ANHOLD         ;RESET AUTO NUMBER CONTROL
83CA   BD   0000G                 JSR    SRQOFF         ;RESET SRQ PENDING STATE
83CD   96   00G                   LDA A  ERRCD,D
83CF   81   60                    CMP A  LASTER,I       ;IN CASE OF BAD ERROR CODE
83D1   23   02                    BLS    SKIPA          ;ALL IS OK
83D3   86   60                    LDA A  LASTER,I       ;PRINT MESSAGE FOR LAST ERROR
83D5   97   00G           SKIPA:  STA A  ERRCDB,D       ;SAVE IN BACK UP AREA
83D7   7F   0000G                 CLR    ERRCD          ;RESET SO INAGTE WILL WORK
83DA   CE   8357                  LDX    MSGTBL-1,I     ;POINTER TO MESSAGE TABLE
83DD   BD   0000G                 JSR    LDBX           ;TRANSLAPTE TO MESSAGE NUMBER
83E0   8D   20                    BSR    NEWMSG         ;CALL MESSAGE WRITER
83E2   7D   0000G                 TST    ANHOLD         ;IF BUFFER IS OPEN NO CR
83E5   26   05                    BNE    ERRSKP
83E7   86   0D                    LDA A  CR,I
83E9   BD   0000G                 JSR    DSPCHR
83EC   DE   00G           ERRSKP: LDX    ZX,D
83EE   FF   0000G                 STX    KEYSTK         ;RESET ANY PENDING KEYS
83F1   FF   0002G                 STX    KEYSTK+2
83F4   FF   0004G                 STX    KEYSTK+4
83F7   FF   0006G                 STX    KEYSTK+6
                          ;*************************************************************
83FA   DF   00G                   STX    CLPTR,D        ;RESET IN CASE OF ABORT *******
                          ;*************************************************************
83FC   4F                 NOERR:  CLR A                 ;RESET SYSTEM VARIABLES
83FD   97   00G                   STA A  ERRCD,D
83FF   97   00G                   STA A  ERRCDB,D
8401   39                         RTS
                          ;
                          ;
8402   4F                 NEWMSG: CLR A
8403   FB   841D                  ADD B  ADDRM+1
8406   B9   841C                  ADC A  ADDRM
8409   97   00G                   STA A  R10,D
840B   D7   01G                   STA B  R10+1,D
840D   DE   00G           MSGLP:  LDX    R10,D          ;GET NEXT OBJECT BYTE
840F   A6   00                    LDA A  0,X
8411   08                         INX
8412   DF   00G                   STX    R10,D
8414   81   F0                    CMP A  240.,I         ;IN COMMAND GROUP
8416   24   06                    BCC    CMMD           ;DO COMMAND
8418   8D   12                    BSR    TEXT           ;TEXT OUTPUT ROUTINE
841A   20   F1                    BRA    MSGLP
```

```
    58
    59          841C   B6CC                ADDRM:  WORD    MSGORG
    60
    61          841E   CE   8474    CMND:   LDX     CMDTBL-240.,I   ;GET DISP TO PSEUDO INSTR
    62          8421   BD   0000G           JSR     LDAX
    63          8424   CE   8495            LDX     CMDORG,I
    64          8427   BD   0000G           JSR     JMPAX           ;CALL COMMAND ROUTINE
    65          842A   20    E1            BRA     MSGLP
```

```
                                        .SBTTL  XXX PRINT TEXT FROM COMPRESSED TABLES
                                ;
                                ;       PRINT TEXT FROM WORD TABLE GIVEN CODE IN ACC-A
                                ;
8420  6F                TEXT: CLR B
8421  48                      ASL A
8422  59                      ROL B
8423  BB  8424               ADD A   ADDRT+1
8426  F9  8483               ADC B   ADDRT
8429  97  01G                 STA A   R11+1,D
842B  D7  00G                 STA B   R11,D
842D  DE  00G                 LDX     R11,D
8430  DF  00G           WRDLP: STX     R11,D
8432  A6  00                  LDA A   0,X
8434  44                      LSR A
8435  44                      LSR A
8436  8D  21                  BSR     OUTPUT
8438  DE  00G                 LDX     R11,D
843A  A6  00                  LDA A   0,X
843C  E6  01                  LDA B   1,X
843E  58                      ASL B
843F  49                      ROL A
8440  58                      ASL B
8441  49                      ROL A
8442  58                      ASL B
8443  49                      ROL A
8444  8D  13                  BSR     OUTPUT
8446  DE  00G                 LDX     R11,D
8448  A6  01                  LDA A   1,X
844A  8D  0D                  BSR     OUTPUT
844C  DE  00G                 LDX     R11,D
844E  08                      INX
844F  08                      INX
8450  6D  00                  TST     0,X
8452  2A  DC                  BPL     WRDLP
8454  86  20                  LDA A   SPACE,I         ;ADD TRAILING SPACE
8456  7E  0000G               JMP     DSPCHR          ;CHARACTER OUTPUT ROUTINE WILL RETURN TO CALLER
                                ;
8459  84  1F            OUTPUT: AND A   31.,I           ;IF NULL SKIP OUT
845B  27  2C                  BEQ     RTS
845D  81  18                  CMP A   L$,I            ;CODE FOR '-'
845F  27  0A                  BEQ     OUTL$
8461  25  10                  BCS     OUTLX           ;CHARACTER
8463  81  1C                  CMP A   L.,I            ; '/' OR ' '
8465  27  08                  BEQ     OUTL.
8467  86  20                  LDA A   SPACE,I
8469  20  0A                  BRA     OUTCRT
846B  86  2D            OUTL$:  LDA A   '-,I
846D  20  06                  BRA     OUTCRT
846F  86  2F            OUTL.:  LDA A   '/,I
8471  20  02                  BRA     OUTCRT
8473  88  40            OUTLX:  ADD A   64.,I           ;CONVERT TO CHARACTER
8475  7E  0000G         OUTCRT: JMP     DSPCHR
                                ;
8478  8637              ADDRT:  .WORD   WRDORG
```

```
                                        .SBTTL  XXX RECALL - RUCURSE THE COMMAND STRING INTERPRITER
                                ;
                                ;       RECALL MESSAGE GENERATION ROUTINES
                                ;
8485    96  01G         RECALL: LDA A   R10+1,D         ;PUSH CURRENT STATUS
8487    36                      PSH A
8488    96  00G                 LDA A   R10,D
848A    36                      PSH A
848B    BD  9402                JSR     NEWMSG          ;NOW CALL EXECUTION LOOP
848E    32                      PUL A                   ;NOW RESET STATUS
848F    97  00G                 STA A   R10,D
8491    32                      PUL A
8492    97  01G                 STA A   R10+1,D
8494    39          RTS:        RTS                     ;AND GO BACK TO CALLER
                                ;
                                ;
8495                CMDORG:                             ;JUMP OFF POINT FOR COMMAND PROCESSORS
```

```
                                        .SBTTL  XXX INLN - IN LINE # MESSAGE
                                ;
                                ;       'IN LINE #' MESSAGE ROUTINE
                                ;
8495    DE  00G     XINLN:  LDX     CLPTR,D         ;IF ZERO I AM LOST
8497    27  FB            BEQ     RTS
8499    EE  07          LDX     PGMLNN,X        ;IF LINE NO. = 0 IT IS IMED EXEC LINE
849B    27  02          BEQ     INLIMD
849D    DF  00G         STX     R12,D           ;SAVE LINE NUMBER TO BE PRINTED
849F    C6  83          LDA B   MINLN,I         ;MESSAGE CONTROL DATA
84A1    7E  8485    INLRCL: JMP     RECALL          ;RECURSE
                                ;
84A4    C6  BF      INLIMD: LDA B   MIMED,I         ;MESSAGE CONTROL DATA
84A6    20  F9          BRA     INLRCL
                                ;
                                ;
                                        .SBTTL  XXX PRLN - PRIOR TO LINE # MESSAGE
                                ;
                                ;       'PRIOR TO LINE #' MESSAGE
                                ;
84A8    DE  00G     XPRLN:  LDX     NLPTR,D         ;NO LINE IF ZERO
84AA    26  08          BNE     PRLNSK          ;IF HAVE LINE TO PRINT NO. FROM NOW
84AC    DE  00G         LDX     CLPTR,D         ;SEE IF FOWARD CHAIN WILL WORK
84AE    27  E4          BEQ     RTS             ;I TRIED MOM
84B0    EE  03          LDX     PGMFP,X         ;LINK UP PROGRAM LIST
84B2    27  E0          BEQ     RTS             ;REAL HARD EVEN
84B4    EE  07      PRLNSK: LDX     PGMLNN,X        ;GET LINE NUMBER
84B6    DF  00G         STX     R12,D
84B8    C6  87          LDA B   MPRLN,I         ;MESSAGE CONTROL DATA
84BA    7E  8485        JMP     RECALL
                                ;
                                ;
                                        .SBTTL  XXX EOF - GET I/O UNIT NUMBER
                                ;
                                ;       GET THE UNIT NUMBER FOR AN EOF ERROR
                                ;
84BD    7F  0000G   XEOF:   CLR     R12
84C0    B6  0000G       LDA A   HLDEOF          ;HOLDING AREA FROM TSTINT
84C3    84  7F          AND A   127.,I          ;FORGET SIGN BIT
84C5    97  01G         STA A   R12+1,D
84C7    39              RTS
```

```
                                                .SBTTL  *** SNTX1 & SNTX2 - SYNTAX ERROR MESSAGES
                                                ;
                                                ;       SYNTAX ERROR ROUTINE - UNLEX MUAST BE CALLED
                                                ;
                                                ;       R13 = NUMBER OF SPACES UNTIL UP ARROW OR SYNTAX MESSAGE
                                                ;       R13+1 - IF ZERO SYNTAX MESSAGE IS BEFORE ARROW
                                                ;
84C8    4F                      XSNTX1: CLR A                   ;RESET SYSTEM VARIABLES
84C9    97      00G                     STA A   R9,D
84CB    97      01G                     STA A   R9+1,D
84CD    CE      FFFFG                   LDX     SCRTCH-1,I      ;ADDRESS OF TRANSL WORK AREA
8400    96      00G                     LDA A   TFLGS1,D        ;NEED TO UNCOMP LINE
8402    28      05                      BMI     SNTXA
8404    08                              INX                     ;GET ADDR OF LINE NO
8405    08                              INX
8406    DF      00G                     STX     R9,D            ;SET UP FOR UNLEX
8408    08                              INX
8409    DF      00G             SNTXA:  STX     R5,D
840B    BD      0000G                   JSR     UNLEX
                                                ;
                                                ;       SYNTAX ERROR WITH LINE IN BUFFER ALREADY
                                                ;
840E    BD      0000G           XSNTX2: JSR     EDTCLR          ;RESET REST OF BUFFER
84E1    C6      C3                      LDA B   #SNTX,I         ;SET UP TO RECALL
84E3    D7      01G                     STA B   R13+1,D         ;SET FLAG AND SAVE MESSAGE CODE
84E5    F7      0000G                   STA B   ANNOLD          ;NO AUTO NUMBER WITH BUFFER OPEN
84E8    B6      0000G                   LDA A   ERCTRB          ;CAN MESSAGE GO BEOFRE ARROW
84EB    81      47                      CMP A   #71.,I          ;SEE IF IT IS WITHIN LINE
84ED    23      05                      BLS     SNIXE           ;OKAY
84EF    86      47                      LDA A   #71.,I
84F1    B7      0000G                   STA A   ERCTRB
84F4    97      00G             SNTXE:  STA A   R13,D           ;SAVE IN WORKING CTR
84F6    80      00                      SUB A   #13.,I
84F8    23      08                      BLS     SNTXJ
84FA    97      00G                     STA A   R13,D           ;SAVE NEW SPACE COUNT
84FC    BD      8485                    JSR     RECALL
84FF    7F      0001G                   CLR     R13+1           ;SAY DONE IN R13 NOW
8502    96      00G             SNTXJ:  LDA A   R13,D           ;CTR MAY BE ZERO
8504    27      0A                      BEQ     SNTXUA          ;TIME FOR UP ARROW
8506    86      20              SNTXLP: LDA A   #SPACE,I        ;SPACES TO MOVE CURSOR
8508    BD      0000G                   JSR     DSPCHR
850B    7A      0000G                   DEC     R13
850E    26      F6                      BNE     SNTXLP
8510    86      7E              SNTXUA: LDA A   #ARROW,I        ;SEND ARROW CODE TO CRT DRIVER
8512    BD      0000G                   JSR     DSPCHR
8515    D6      01G                     LDA B   R13+1,D         ;TEST CODE (ZERO OR MSNTX CODE)
8517    27      0A                      BEQ     SNTXZ           ;ALREADY DONE
8519    86      20                      LDA A   #SPACE,I
851B    BD      0000G                   JSR     DSPCHR
851E    C6      C3                      LDA B   #MSNTX,I
8520    BD      8485                    JSR     RECALL          ;PRINT MESSAGE NOW
8523    4F              SNTXZ:  CLR A
8524    F6      0000G                   LDA B   ERCTRB          ;FIND OUT WHERE TO PUT CURSOR
8527    FB      8537                    ADD B   #BFR+1
852A    B9      8536                    ADC A   #BFR
852D    97      00G                     STA A   EDTPTR,D        ;CURSOR LOCATION AT RECALL TIME
852F    D7      01G                     STA B   EDTPTR+1,D
```

XXX SNTX1 & SNTX2 - SYNTAX ERROR MESSAGES

```
                                        .IF     EQ,1
                                        CLR     R12             ; TEMP FIX FOR BROKEN PARSER OR LEXER
                                        LDA A   ERCTRA
                                        STA A   R12+1,D
                                        JSR     XMMBR
                                        CLR     R12
                                        LDA A   ERCTRB
                                        STA A   R12+1,D
                                        JSR     XMMBR
                                        .ENDC
8531    86      BJ                      LDA     RCLKEY.I        ;FAKE EDITOR CMJT
8533    7E      0000G                   JMP     TYPIN
                                ;
8536    0000G           ABER:   .WORD   EDTBER
```

```
                                        .SBTTL  XXX MSGN - PRINT MESSAGE NUMBER AND RETURN TO CALLER
                                ;
                                ;       PRINT '- MESSAGE NUMBER #'
                                ;
8538    7F    0000G     XMSGN:  CLR     R12             ;SET UP WORK AREA
8538    96      00G             LDA A   ERRCOB,D
853D    97      01G             STA A   R12+1,D
853F    C6      88              LDA B   #MSGN.I         ;COMMAND STRING
8541    32                      PUL A                   ;FORGET CALLERS RETURN POINT
8542    32                      PUL A
8543    7E    8485              JMP     RECALL
```

```
                                        .SBTTL  *** NMBR - PRINT THE NUMBER IN R12
                                ;
                                ;       PRINT THE NUMBER IN R12
                                ;
8546  DE  00G   XNMBR:  LDX   R12,0      ;GET DATA
8548  BD  0000G        JSR   INAGTE     ;USE SYSTEM CONVERSION ROUTINE
8548  CE  0001G        LDX   R16+1,I    ;GET ADDR OF FIRST BYTE OF OUTPUT
854E  DF  00G   NMBRLP: STX   R12,0
8550  A6  00           LDA A 0,X
8552  BD  0000G        JSR   DSPCHR
8555  DE  00G          LDX   R12,0
8557  08               INX
8558  A6  00           LDA A 0,X        ;ANY MORE DATA
855A  26  F2           BNE   NMBRLP
855C  86  20           LDA A SPACE,I    ;SPACES ARE FREE
855E  7E  0000G        JMP   DSPCHR
```

```
                                        .SBTTL  *** EXIT COMMAND
                                ;
                                ;
8561    32              XEXIT:  PUL A                   ;FORGET WHO CALLED ME
8562    32                      PUL A
8563    39                      RTS
                                ;
                                ;
                                        .SBTTL  *** COMMAND JUMP TABLES
                                ;
                                ;       COMMAND TABLE
                                ;
8564                    CMDTBL: CMD     INLN
8565                            CMD     PRLN
8566                            CMD     EOF
8567                            CMD     SNTX1
8568                            CMD     SNTX2
8569                            CMD     MSGN
856A                            CMD     NMBR
8568                            CMD     EXIT
```

```
                                .SBTTL  xxx MESSAGE TABLES
                        ;
                        ;       MESSAGE DIFENATION TABLES
                        ;
856C            MSGORG: MSG     1,SIZE,ERROR,CINLN,CMSGN
8570                    MSG     2,STOP,CINLN,CPRLN,CEXIT
8574                    MSG     3,PGM,ABORT,CINLN,CEXIT
8578                    MSG     4,CSNTX1,CEXIT
857A                    MSG     5,CSNTX2,CEXIT
857C                    MSG     6,INVLD,LINE,NMBR,CMSGN
8580                    MSG     7,SHAPE,ERROR,CINLN,CMSGN
8584                    MSG     8,DIM,ERROR,CINLN,CMSGN
8588                    MSG     9,FN,UNDF,CINLN,CMSGN
858C                    MSG     10,INVLD,CMD,ARG,CINLN,CMSGN
8591                    MSG     11,INVLD,FN,ARG,CINLN,CMSGN
8596                    MSG     12,CALL,INVLD,CINLN,CMSGN
859A                    MSG     13,DATA,STMT,INVLD,CINLN,CMSGN
859F                    MSG     14,LINE,NMBR,REQ,CINLN,CMSGN
85A4                    MSG     15,UNDF,VAR,CINLN,CMSGN
85A8                    MSG     16,EXTND,FN,ROM,REQ,CINLN,CMSGN
85AE                    MSG     17,PGM,IS,SECRET,CINLN,CMSGN
85B3                    MSG     18,MEM,IS,FULL,CINLN,CMSGN
85B8                    MSG     19,PGM,INTRPT,CPRLN,CEXIT
85BC                    MSG     20,SYS,ERROR,CINLN,CMSGN
85C0                    MSG     21,NO,SRQ,ON,UNIT,CINLN,CMSGN
85C6                    MSG     22,NO,EXT,ON,UNIT,CINLN,CMSGN
85CC                    MSG     23,EOF,ON,UNIT,CEOF,CNMBR,CINLN,CMSGN
8503                    MSG     24,STMT,IS,TOO,LONG,CINLN,CMSGN
8509                    MSG     25,INVLD,LINE,NMBR,CINLN,CMSGN
85DE                    MSG     26,MT,FILE,NOT,FND,CINLN,CMSGN
85E4                    MSG     27,MT,READ,ERROR,CINLN,CMSGN
85E9                    MSG     28,MT,ERROR,CINLN,CMSGN
85E0                    MSG     29,MT,IS,WRTPR,CINLN,CMSGN
85F2                    MSG     30,NO,PGM,FND,CINLN,CMSGN
85F7                    MSG     31,MT,CART,REQ,CINLN,CMSGN
85FC                    MSG     32,INVLD,MT,FMT,CINLN,CMSGN
8601                    MSG     33,INVLD,OPR,ON,AN,OPEN,FILE,CINLN,CMSGN
8609                    MSG     34,INVLD,IO,OPR,CINLN,CMSGN
860E                    MSG     35,GPBUS,IO,ERROR,CINLN,CMSGN
8613                    MSG     36,PRT,FMT,ERROR,CINLN,CMSGN
8618                    MSG     37,FIELD,OVERFL,IN,PRT,FMT,CINLN,CMSGN
861F                    MSG     INLN,IN,LINE,CNMBR,CEXIT
8623                    MSG     PRLN,PRIORT,LINE,CNMBR,CEXIT
8627                    MSG     MSGN,MSG,NMBR,CNMBR,CEXIT
862B                    MSG     IMED,IN,IMED,LINE,CEXIT
862F                    MSG     SNTX,SYNTAX,ERROR,CEXIT
8632                    MSG     ROM1,ERROR,CINLN,CPRLN,CMSGN
8636                    MSG     ROM2,CEXIT
00CB            LENMSG  =       .-MSGORG                ;LENGTH OF TABLE
```

```
                                        .SBTTL  XXX WORD TABLES
                                ;
                                ;       GENERATE THE SYMBOLS FOR RAD 32 CHARACTERS
                                ;
                        0001    CTR     =       1
                                        .IRPC   CHR,ABCDEFGHIJKLMNOPQRSTUVWXYZ$.
                                L'CHR   =       CTR
                                CTR     =       CTR+1
                                        .ENDM
                        001D    L       =       CTR             ;SPACE CHARACTER
                                ;
                                ;       GENERATE THE WORD TABLES
                                ;
                8637    WRDORG: WRD     ABORT,<<ABORTED>>
                8639            WRD     AN,<<AN>>
                863F            WRD     ARG,<<ARGUMENT>>
                8645            WRD     CALL,<<CALL NAME>>
                8648            WRD     CART,<<CARTRIDGE>>
                8651            WRD     CMD,<<COMMAND>>
                8652            WRD     DATA,<<DATA>>
                8658            WRD     DIM,<<DIMENSIONSSUBSCRIPT>>
                8669            WRD     EOF,<<EOF>>
                8668            WRD     ERROR,<<ERROR>>
                866F            WRD     EXT,<<EXTERNAL>>
                8675            WRD     EXTND,<<EXTENDED>>
                8678            WRD     FIELD,<<FIELD>>
                867F            WRD     FILE,<<FILE>>
                8683            WRD     FMT,<<FORMAT>>
                8687            WRD     FN,<<FUNCTION>>
                8680            WRD     FND,<<FOUND>>
                8691            WRD     FULL,<<FULL>>
                8695            WRD     GPBUS,<<GP INTERFACE BUS>>
                86A1            WRD     IMED,<<IMMEDIATE>>
                86A7            WRD     IN,<<IN>>
                86A9            WRD     INTRPT,<<INTERRUPTED>>
                8681            WRD     INVLD,<<INVALID>>
                8687            WRD     IO,<<I.O>>
                8689            WRD     IS,<<IS>>
                8688            WRD     LINE,<<LINE>>
                868F            WRD     LONG,<<LONG>>
                86C3            WRD     MEM,<<MEMORY>>
                86C7            WRD     MSG,<<S MESSAGE>>
                86CD            WRD     MT,<<MAG TAPE>>
                8603            WRD     NMBR,<<NUMBER>>
                8607            WRD     NO,<<NO>>
                8609            WRD     NOT,<<NOT>>
                8608            WRD     ON,<<ON>>
                8600            WRD     OPEN,<<OPEN>>
                86E1            WRD     OPR,<<OPERATION>>
                86E7            WRD     OVERFL,<<OVERFLOW>>
                86ED            WRD     PGM,<<PROGRAM>>
                86F3            WRD     PRT,<<PRINT>>
                86F7            WRD     PRIORI,<<PRIOR TO>>
                86FD            WRD     READ,<<READ>>
                8701            WRD     REQ,<<REQUIRED>>
                8707            WRD     ROM,<<ROM>>
                8709            WRD     SECRET,<<SECRET>>
```

```
XXX WORD TABLES

                8700                            WRD     SHAPE,<<SHAPE>>
                8711                            WRD     SIZE,<<SIZE>>
                8715                            WRD     SRQ,<<SRQ>
                8717                            WRD     STAT,<<ST.   ENT>>
                8710                            WRD     STOP,<<STOP>>
                8721                            WRD     SYNTAX,<<SYNTAX>>
                8725                            WRD     SYS,<<SYSTEM>>
                8729                            WRD     TOO,<<TOO>>
                8728                            WRD     UNDF,<<UNDEFINED>>
                8731                            WRD     UNIT,<<UNIT>>
                8735                            WRD     VAR,<<VARIABLE>>
                8738                            WRD     WRTPR,<<WRITE PROTECTED>>
                8745    80                      .BYTE   128.            ;END OF TABLE MARKER
                        010F            LENWRD  =       .-WRDORG        ;LENGTH OF TABLE
                                        ;
                                        ;
                                        ;
                0001'                   .END
```

SYMBOL TABLE

```
ABFR    8536        ABORT = 0000        ADDRM   841C        ADDRT   8483        AN    = 0003
ANHOLD= XXXXXX G    ARG   = 0004        ARROW = 002E        ASX   = XXXXXX G    CALL  = 0007
CART  = 000A        CEOF  = 00F2        CEXIT = 00F7        CINLN = 00F0        CLPTR = XXXXXX G
CMD   = 0000        CMDORG  8495        CMDTBL  8564        CMMD    841E        CMSGN = 00F5
CNMBR = 00F6        CPRLN = 00F1        CR    = 0000        CSNTX1= 00F3        CSNTX2= 00F4
CTR   = 0003        DATA  = 0010        DIM   = 0012        DSPCHR= XXXXXX G    EDTBFR= XXXXXX G
EDTCLR= XXXXXX G    EDTPTR= XXXXXX G    EOF   = 0019        ERAPP1= 000E G      ERAPP2= 000F G
ERARC = 0018 G      ERASGN= 0016 G      ERATSN= 0042 G      ERBCNU= 004E G      ERBDPU= 0050 G
ERBMDU= 0040 G      ERBPNU= 004C G      ERBRK = 0028 G      ERCARC= 001A G      ERCHAR= 005E G
ERCLAI= 000C G      ERCNSF= 0020 G      ERCTRA= XXXXXX G    ERCTRB= XXXXXX G    ERDAGN= 0056 G
ERDATM= 0051 G      ERDEOR= 0063 G      ERDET = 001F G      ERDIM = 0009 G      ERDOMN= 0018 G
EREOFN= 0014 G      EREOM = 0036 G      EREXEC= 0044 G      EREXP = 0004 G      ERFBFR= 0031 G
ERFILE= 003E G      ERFIXN= 005F G      ERFLDO= 0057 G      ERFNFO= 0034 G      ERFORA= 0013 G
ERFPDV= 0002 G      ERFPOV= 0001 G      ERFUZZ= 0010 G      ERFXOV= 0060 G      ERIEC = 0049 G
ERINFS= 0048 G      ERINTR= 0058 G      ERINU = 004A G      ERIOE = 0045 G      ERISGN= 0054 G
ERISTS= 0055 G      ERLNNF= 0033 G      ERLNNO= 0007 G      ERLOG = 0017 G      ERLOLD= 0032 G
ERLRSU= 004E G      ERMARK= 0012 G      ERMEMT= 003A G      ERMILA= 0037 G      ERMOPN= 0030 G
ERMTRD= 0035 G      ERMUL = 001E G      ERNCN = 0021 G      ERNCRT= 0039 G      ERNDT = 0022 G
ERNFST= 0047 G      ERNIMX= 0023 G      ERNIOO= 0043 G      ERNOFN= 000B G      ERNOLD= 0038 G
ERNON1= 0029 G      ERNON2= 002A G      ERNON3= 002B G      ERNONW= 002C G      ERNONS= 002D G
ERNON6= 002E G      ERNON7= 002F G      ERNON8= 0030 G      ERNSEP= 0041 G      ERNXFN= 0025 G
ERNXTA= 0013 G      EROFA = 0013 G      EROVRD= 003C G      ERQUOT= 0046 G      ERRCD = XXXXXX G
ERRCOR= XXXXXX G    ERRCTR= 0061        ERREAD= 003F G      ERREN = 0011 G      ERREP = 001C G
ERROM1= 0059 G      ERROM2= 005A G      ERROR = 001A        ERRSKP  83EC        ERSEC = 0026 G
ERSFE = 0019 G      ERSHAP= 0008 G      ERSOR = 0006 G      ERSTGL= 0040 G      ERSTOP= 0058 G
ERSUBC= 0008 G      ERSYNX= 0050 G      ERTABO= 0052 G      ERTERM= 005C G      ERTMDS= 0048 G
ERTRNG= 0005 G      ERUNDF= 0024 G      ERUP  = 0003 G      ERUPN = 0016 G      ERVAL = 001D G
ERWBYT= 0000 G      ERWRT = 0038 G      ERWSFL= 0027 G      EXT   = 001C        EXTND = 001F
FIELD = 0022        FILE  = 0024        FIX1  = XXXXXX G    FMT   = 0026        FN    = 0028
FND   = 0028        FULL  = 002D        GLBFLG= XXXXXX G    GPBUS = 002F        HLDEOF= XXXXXX G
IMED  = 0035        IN    = 0038        INAGTE= XXXXXX G    INLIMD  84A4        INLRCL  84A1
INTRPT= 0039        INVLD = 0030        IO    = 0040        IS    = 0041        JMPAX = XXXXXX G
KEYSTK= XXXXXX G    L     = 001D        LA    = 0001        LASTER= 0060        LB    = 0002
LC    = 0003        LD    = 0004        LDAX  = XXXXXX G    LDBX  = XXXXXX G    LDXX  = XXXXXX G
LE    = 0005        LENMSG= 00C8        LENWRD= 010F        LF    = 0006        LG    = 0007
LH    = 0008        LI    = 0009        LINE  = 0042        LJ    = 000A        LK    = 000B
LL    = 000C        LM    = 0000        LN    = 000E        LO    = 000F        LONG  = 0044
LP    = 0010        LQ    = 0011        LR    = 0012        LS    = 0013        LT    = 0014
LU    = 0015        LV    = 0016        LW    = 0017        LX    = 0018        LY    = 0019
LZ    = 001A        L$    = 001B        L.    = 001C        MEM   = 0046        MIMED = 008F
MINLN = 0083        MMSGN = 0088        MPRLN = 0087        MROM1 = 00C6        MROM2 = 00C8
MSG   = 0048        MSGLP   8400        MSGORG  8F6C        MSGTBL  8358        MSNTX = 00C3
MT    = 0048        M1    = 0000        M10   = 0020        M11   = 0025        M12   = 0029
M13   = 002E        M14   = 0033        M15   = 0038        M16   = 003C        M17   = 0042
M18   = 0047        M19   = 004C        M2    = 0004        M20   = 0050        M21   = 0054
M22   = 005A        M23   = 0060        M24   = 0067        M25   = 006D        M26   = 0072
M27   = 0078        M28   = 007D        M29   = 0081        M3    = 0008        M30   = 0086
M31   = 0088        M32   = 0090        M33   = 0095        M34   = 0090        M35   = 00A2
M36   = 00A7        M37   = 00AC        M4    = 00CC        M5    = 000E        M6    = 0010
M7    = 0014        M8    = 0018        M9    = 001C        NEWMSG  8402        NLPTR = XXXXXX G
NMBR  = 004E        NMBRLP  854E        NO    = 0050        NOERR   83FC        NOT   = 0051
ON    = 0052        OPEN  = 0053        OPR   = 0055        OUTCRT  8480        OUTLX   847E
OUTL$   8476        OUTL    847A        OUTPUT  8464        OVERFL= 0058        PACKED= 0000
PGM   = 0058        PGMFP = 0003        PGMLNN= 0007        PRIORT= 0060        PRLNSK  8484
PRT   = 005E        PRTERR  8388 G      RCLKEY= 0083        READ  = 0063        RECALL  8485
REQ   = 0065        ROM   = 0068        RTS     8494        R10   = XXXXXX G    R11   = XXXXXX G
R12   = XXXXXX G    R13   = XXXXXX G    R16   = XXXXXX G    R5    = XXXXXX G    R9    = XXXXXX G
```

```
PRTERR ERROR CODE LISTING WITH RT-11 MMAC  VM02-10  14-OCT-76 01:38:57 PAGE 13+
SYMBOL TABLE

SCRTCH= xxxxxx G    SECRET=  0069      SHAPE =  0068      SIZE  =  0060      SIZERR=  0006 G
SKIPA    8705       SNTXA    8409      SNTXE    84E4      SNTXJ    8502      SNTXLP   8506
SNTXUA   8510       SNTXZ    8523      SPACE =  0020      SRQ   =  006F      SRQOFF=  xxxxxx G
STMT  =  0070       STOP  =  0073      SYNTAX=  0075      SYS   =  0077      TEXT     842C
TFLGS1= xxxxxx G    TOG   =  0079      TYPIN = xxxxxx G   UNDF  =  002A      UNLT  =  0070
UNLEX = xxxxxx G    VAR   =  007F      WRDLP    8438      WRDORG   8637      WRTPR =  0082
XEOF     8480       XEXIT    8561      XINLN    8495      XMSGN    8538      XNMBR    8546
XPRLN    8408       XSNTX1   84C8      XSNTX2   840E      ZX    = xxxxxx G

. ABS.    8746    00
          0000    01
ERRORS DETECTED: 0    WARNINGS POSTED: 0    FREE CORE: 1668. WORDS
.SY:PRTERR/C(DK1:SEICLI,PRTERR
```

```
PRTERR ERROR CODE LISTING WITH RT-11 MMAC  VM02-10   14-OCT-76 01:38:57 PAGE S-1
CROSS REFERENCE TABLE (CREF V01-03 )


        1-29#   11-13   11-14   11-15   11-16   11-17   11-18   11-19   11-20   12-5    12-6    12-7    12-8    12-9
        12-10   12-11   12-12   12-13   12-14   12-15   12-16   12-17   12-18   12-19   12-20   12-21   12-22   12-23
        12-24   12-25   12-26   12-27   12-28   12-29   12-30   12-31   12-32   12-33   12-34   12-35   12-36   12-37
        12-38   12-39   12-40   12-41   12-42   12-43   12-44   12-45   12-46   12-47   12-48   12-49   13-14   13-15
        13-16   13-17   13-18   13-19   13-20   13-21   13-22   13-23   13-24   13-25   13-26   13-27   13-28   13-29
        13-30   13-31   13-32   13-33   13-34   13-35   13-36   13-37   13-38   13-39   13-40   13-41   13-42   13-43
        13-44   13-45   13-46   13-47   13-48   13-49   13-50   13-51   13-52   13-53   13-54   13-55   13-56   13-57
        13-58   13-59   13-60   13-61   13-62   13-63   13-64   13-65   13-66   13-67   13-68   13-69   13-71
ASX     3-8#
ABER    8-54    8-56    8-71#
ABORT   12-7    13-14#
ADDR    4-46    4-47    4-59#
ADDRT   5-8     5-9     5-55#
AN      12-37   13-15#
ANHOLD  3-11#   4-16*   4-27    8-26*
ARG     12-14   12-15   13-16#
ARROW   3-21#   8-44
CALL    12-16   13-17#
CART    12-35   13-18#
CEOF    11-15#  12-27
CEXIT   11-20#  12-6    12-7    12-8    12-9    12-23   12-42   12-43   12-44   12-45   12-46   12-48
CINLN   11-13#  12-5    12-6    12-7    12-11   12-12   12-13   12-14   12-15   12-16   12-17   12-18   12-19   12-20
        12-21   12-22   12-24   12-25   12-26   12-27   12-28   12-29   12-30   12-31   12-32   12-33   12-34   12-35
        12-36   12-37   12-38   12-39   12-40   12-41   12-47
CLPTR   3-5#    4-37*   7-5     7-23
CMD     12-14   13-19#
CMDORG  4-63    6-17#   11-13   11-14   11-15   11-16   11-17   11-18   11-19   11-20
CMDTBL  4-61    11-13   11-13#  11-14   11-15   11-16   11-17   11-18   11-19   11-20
CMMD    4-55    4-61#
CMSGN   11-18#  12-5    12-10   12-11   12-12   12-13   12-14   12-15   12-16   12-17   12-18   12-19   12-20   12-21
        12-22   12-24   12-25   12-26   12-27   12-28   12-29   12-30   12-31   12-32   12-33   12-34   12-35   12-36
        12-37   12-38   12-39   12-40   12-41   12-47
CNMBR   11-19#  12-27   12-42   12-43   12-44
CPRLN   11-14#  12-6    12-23   12-47
CR      3-19#   4-14    4-29
CSNTX1  11-16#  12-8
CSNTX2  11-17#  12-9
CTR     13-5#   13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9
        13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9
        13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9
        13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9    13-9
        13-9    13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#
        13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#   13-9#
        13-9#   13-10   13-14   13-14   13-14   13-14   13-14   13-14   13-14   13-14   13-14   13-14   13-14   13-14
        13-14   13-14   13-14   13-14   13-14#  13-14#  13-14#  13-14#  13-14#  13-14#  13-14#  13-14#  13-14#  13-14#
        13-15   13-15   13-15   13-15   13-15   13-15   13-15#  13-15#  13-15#  13-16   13-16   13-16   13-16   13-16
        13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16   13-16#
        13-16#  13-16#  13-16#  13-16#  13-16#  13-16#  13-16#  13-16#  13-16#  13-16#  13-17   13-17   13-17   13-17
        13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17   13-17
        13-17   13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#  13-17#
        13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18   13-18
        13-18   13-18   13-18   13-18   13-18   13-18#  13-18#  13-18#  13-18#  13-18#  13-18#  13-18#  13-18#  13-18#
        13-18#  13-18#  13-18#  13-18#  13-19   13-19   13-19   13-19   13-19   13-19   13-19   13-19   13-19   13-19
        13-19   13-19   13-19   13-19   13-19   13-19   13-19#  13-19#  13-19#  13-19#  13-19#  13-19#  13-19#  13-19#
        13-19#  13-19#  13-20   13-20   13-20   13-20   13-20   13-20   13-20   13-20   13-20   13-20   13-20#  13-20#
        13-20#  13-20#  13-20#  13-20#  13-21   13-21   13-21   13-21   13-21   13-21   13-21   13-21   13-21   13-21
```

13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21
13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21	13-21
13-21	13-21	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#
13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#	13-21#
13-22	13-22	13-22	13-22	13-22	13-22	13-22	13-22#	13-22#	13-22#	13-22#	13-22#	13-23	13-23
13-23	13-23	13-23	13-23	13-23	13-23	13-23	13-23	13-23	13-23	13-23#	13-23#	13-23#	13-23#
13-23#	13-23#	13-23#	13-24	13-24	13-24	13-24	13-24	13-24	13-24	13-24	13-24	13-24	13-24
13-24	13-24	13-24	13-24	13-24	13-24	13-24	13-24#	13-24#	13-24#	13-24#	13-24#	13-24#	13-24#
13-24#	13-24#	13-24#	13-24#	13-25	13-25	13-25	13-25	13-25	13-25	13-25	13-25	13-25	13-25
13-25	13-25	13-25	13-25	13-25	13-25	13-25	13-25	13-25#	13-25#	13-25#	13-25#	13-25#	13-25#
13-25#	13-25#	13-25#	13-25#	13-25#	13-26	13-26	13-26	13-26	13-26	13-26	13-26	13-26	13-26
13-26	13-26	13-26	13-26#	13-26#	13-26#	13-26#	13-26#	13-26#	13-26#	13-27	13-27	13-27	13-27
13-27	13-27	13-27	13-27	13-27	13-27	13-27#	13-27#	13-27#	13-27#	13-27#	13-27#	13-28	13-28
13-28	13-28	13-28	13-28	13-28	13-28	13-28	13-28	13-28	13-28	13-28	13-28#	13-28#	13-28#
13-28#	13-28#	13-28#	13-28#	13-28#	13-28#	13-29	13-29	13-29	13-29	13-29	13-29	13-29	13-29
13-29	13-29	13-29	13-29	13-29	13-29	13-29	13-29	13-29	13-29	13-29#	13-29#	13-29#	13-29#
13-29#	13-29#	13-29#	13-29#	13-29#	13-29#	13-29#	13-30	13-30	13-30	13-30	13-30	13-30	13-30
13-30	13-30	13-30	13-30	13-30	13-30#	13-30#	13-30#	13-30#	13-30#	13-30#	13-30#	13-31	13-31
13-31	13-31	13-31	13-31	13-31	13-31	13-31	13-31	13-31#	13-31#	13-31#	13-31#	13-31#	13-31#
13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32
13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32	13-32
13-32	13-32	13-32	13-32	13-32	13-32	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#
13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#	13-32#
13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33	13-33
13-33	13-33	13-33	13-33	13-33	13-33#	13-33#	13-33#	13-33#	13-33#	13-33#	13-33#	13-33#	13-33#
13-33#	13-33#	13-33#	13-33#	13-34	13-34	13-34	13-34	13-34	13-34	13-34#	13-34#	13-34#	13-35
13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35
13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35	13-35#	13-35#	13-35#	13-35#	13-35#
13-35#	13-35#	13-35#	13-35#	13-35#	13-35#	13-35#	13-35#	13-35#	13-35#	13-36	13-36	13-36	13-36
13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36	13-36#	13-36#
13-36#	13-36#	13-36#	13-36#	13-36#	13-36#	13-36#	13-36#	13-37	13-37	13-37	13-37	13-37	13-37
13-37	13-37#	13-37#	13-37#	13-37#	13-37#	13-38	13-38	13-38	13-38	13-38	13-38	13-38#	13-38#
13-38#	13-39	13-39	13-39	13-39	13-39	13-39	13-39	13-39	13-39	13-39	13-39#	13-39#	13-39#
13-39#	13-39#	13-39#	13-40	13-40	13-40	13-40	13-40	13-40	13-40	13-40	13-40	13-40	13-40#
13-40#	13-40#	13-40#	13-40#	13-40#	13-41	13-41	13-41	13-41	13-41	13-41	13-41	13-41	13-41
13-41	13-41	13-41	13-41	13-41#	13-41#	13-41#	13-41#	13-41#	13-41#	13-41#	13-41#	13-41#	13-42
13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42	13-42
13-42	13-42	13-42	13-42	13-42#	13-42#	13-42#	13-42#	13-42#	13-42#	13-42#	13-42#	13-42#	13-42#
13-42#	13-42#	13-42#	13-43	13-43	13-43	13-43	13-43	13-43	13-43	13-43	13-43	13-43	13-43
13-43	13-43	13-43	13-43	13-43	13-43	13-43	13-43#	13-43#	13-43#	13-43#	13-43#	13-43#	13-43#
13-43#	13-43#	13-43#	13-43#	13-44	13-44	13-44	13-44	13-44	13-44	13-44	13-44	13-44	13-44
13-44	13-44	13-44	13-44#	13-44#	13-44#	13-44#	13-44#	13-44#	13-44#	13-44#	13-44#	13-45	13-45
13-45	13-45	13-45	13-45	13-45#	13-45#	13-45#	13-46	13-46	13-46	13-46	13-46	13-46	13-46
13-46#	13-46#	13-46#	13-46#	13-46#	13-47	13-47	13-47	13-47	13-47	13-47	13-47#	13-47#	13-47#
13-48	13-48	13-48	13-48	13-48	13-48	13-48	13-48	13-48	13-48	13-48#	13-48#	13-48#	13-48#
13-48#	13-48#	13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49
13-49	13-49	13-49	13-49	13-49	13-49	13-49	13-49#	13-49#	13-49#	13-49#	13-49#	13-49#	13-49#
13-49#	13-49#	13-49#	13-49#	13-49#	13-49#	13-50	13-50	13-50	13-50	13-50	13-50	13-50	13-50
13-50	13-50	13-50	13-50	13-50	13-50	13-50	13-50	13-50	13-50	13-50#	13-50#	13-50#	13-50#
13-50#	13-50#	13-50#	13-50#	13-50#	13-50#	13-50#	13-51	13-51	13-51	13-51	13-51	13-51	13-51
13-51	13-51	13-51	13-51	13-51	13-51	13-51	13-51	13-51	13-51#	13-51#	13-51#	13-51#	13-51#
13-51#	13-51#	13-51#	13-51#	13-51#	13-52	13-52	13-52	13-52	13-52	13-52	13-52	13-52	13-52
13-52	13-52	13-52	13-52#	13-52#	13-52#	13-52#	13-52#	13-52#	13-52#	13-53	13-53	13-53	13-53
13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53	13-53
13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-53#	13-54	13-54
13-54	13-54	13-54	13-54	13-54	13-54	13-54	13-54#	13-54#	13-54#	13-54#	13-54#	13-54#	13-55
13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55	13-55

CROSS REFERENCE TABLE (CREF V01-03)

13-55 13-55 13-55 13-55# 13-55# 13-55# 13-55# 13-55# 13-55# 13-55# 13-55# 13-55# 13-55# 13-55#
13-56 13-56 13-56 13-56 13-56 13-56 13-56 13-56# 13-56# 13-56# 13-56# 13-56# 13-57 13-57
13-57 13-57 13-57 13-57 13-57 13-57 13-57 13-57 13-57 13-57 13-57 13-57# 13-57# 13-57#
13-57# 13-57# 13-57# 13-57# 13-57# 13-57# 13-58 13-58 13-58 13-58 13-58 13-58 13-58 13-58
13-58 13-58 13-58 13-58 13-58# 13-58# 13-58# 13-58# 13-58# 13-58# 13-58# 13-59 13-59 13-59
13-59 13-59 13-59 13-59 13-59 13-59 13-59 13-59# 13-59# 13-59# 13-59# 13-59# 13-59# 13-60
13-60 13-60 13-60 13-60 13-60 13-60 13-60# 13-60# 13-60# 13-60# 13-60# 13-61 13-61 13-61
13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61 13-61
13-61 13-61 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61# 13-61#
13-61# 13-62 13-62 13-62 13-62 13-62 13-62 13-62 13-62 13-62 13-62 13-62# 13-62# 13-62#
13-62# 13-62# 13-62# 13-63 13-63 13-63 13-63 13-63 13-63 13-63 13-63 13-63 13-63 13-63
13-63 13-63 13-63# 13-63# 13-63# 13-63# 13-63# 13-63# 13-63# 13-63# 13-63# 13-64 13-64 13-64
13-64 13-64 13-64 13-64 13-64 13-64 13-64 13-64 13-64 13-64 13-64# 13-64# 13-64# 13-64#
13-64# 13-64# 13-64# 13-64# 13-64# 13-65 13-65 13-65 13-65 13-65 13-65 13-65 13-65# 13-65#
13-65# 13-65# 13-65# 13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66
13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66 13-66# 13-66# 13-66# 13-66# 13-66# 13-66#
13-66# 13-66# 13-66# 13-66# 13-66# 13-66# 13-66# 13-67 13-67 13-67 13-67 13-67 13-67 13-67
13-67 13-67 13-67 13-67# 13-67# 13-67# 13-67# 13-67# 13-67# 13-68 13-68 13-68 13-68 13-68
13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68 13-68#
13-68# 13-68# 13-68# 13-68# 13-68# 13-68# 13-68# 13-68# 13-68# 13-68# 13-69 13-69 13-69 13-69
13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69
13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69 13-69#
13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69# 13-69#
13-69# 13-69# 13-69# 13-69# 13-69# 13-69#

DATA 12-17 13-2
DIM 12-12 13-21#
DSPCHR 3-4# 4-15 4-30 5-37 5-53 8-41 8-45 8-49 10-10 10-16
EDTBFR 3-6# 8-21
EDTCLR 3-6# 8-23
EDTPTR 3-6# 8-26# 8-57#
EOF 12-27 13-22#
ERAPP1 2-43# 2-43#
ERAPP2 2-45# 2-45#
ERARC 2-81# 2-81#
ERASGN 2-68# 2-68#
ERATSN 2-182# 2-182#
ERBCMU 2-211# 2-211#
ERBDPU 2-213# 2-213#
ERBMDU 2-206# 2-206#
ERBPNU 2-204# 2-204#
ERBRK 2-124# 2-124#
ERCARC 2-86# 2-86#
ERCHAR 2-244# 2-244#
ERCLAT 2-39# 2-39#
ERCNSF 2-99# 2-99#
ERCTRA 3-3#
ERCTRB 3-3# 8-27 8-31# 8-53
ERDAGN 2-225# 2-225#
ERDATM 2-215# 2-215#
ERDEOR 2-219# 2-219#
ERDET 2-96# 2-96#
ERDIM 2-27# 2-27#
ERDMMN 2-87# 2-87#
EREOFN 2-66# 2-66#
EREOM 2-157# 2-157#
EREXEC 2-188# 2-188#
EREXP 2-13# 2-13#

CROSS REFERENCE TABLE (CREF V01-03)

ERFBER	2-142#	2-142#													
ERFILE	2-175#	2-175#													
ERFIXN	2-247#	2-247#													
ERFLDO	2-227#	2-227#													
ERFNFD	2-149#	2-149#													
ERFORA	2-60#	2-60#													
ERFPDV	2-8#	2-8#													
ERFPOV	2-5#	2-5#													
ERFUZZ	2-47#	2-47#													
ERFXOV	2-250#	2-250#													
ERIFC	2-198#	2-198#													
ERINFS	2-195#	2-195#													
ERINTR	2-229#	2-229#													
ERINU	2-200#	2-200#													
ERIOE	2-189#	2-189#													
ERISGN	2-221#	2-221#													
ERISTS	2-223#	2-223#													
ERLNNF	2-146#	2-146#													
ERLNNO	2-22#	2-22#													
ERLOG	2-78#	2-78#													
ERLOLD	2-144#	2-144#													
ERLRSU	2-208#	2-208#													
ERMARK	2-55#	2-55#													
ERMFMT	2-166#	2-166#													
ERMILA	2-160#	2-160#													
ERMOPN	2-173#	2-173#													
ERMTRD	2-152#	2-152#													
ERMUL	2-94#	2-94#													
ERNCN	2-101#	2-101#													
ERNCRT	2-163#	2-163#													
ERNDT	2-102#	2-102#													
ERNFST	2-193#	2-193#													
ERNIMX	2-106#	2-106#													
ERNLOD	2-185#	2-185#													
ERNOFN	2-37#	2-37#													
ERNOLD	2-169#	2-169#													
ERNON1	2-126#	2-126#													
ERNON2	2-128#	2-128#													
ERNON3	2-130#	2-130#													
ERNON4	2-132#	2-132#													
ERNON5	2-134#	2-134#													
ERNON6	2-136#	2-136#													
ERNON7	2-138#	2-138#													
ERNON8	2-140#	2-140#													
ERNSEP	2-180#	2-180#													
ERNXFN	2-114#	2-114#													
ERNXTA	2-56#	2-56#													
EROFA	2-63#	2-63#													
EROVRD	2-171#	2-171#													
ERQUOT	2-191#	2-191#													
ERRCD	3-3#	4-9	4-18	4-23%	4-40%										
ERRCDB	3-3#	4-22%	4-41%	9-6											
ERRCTR	2-4#	2-5	2-5	2-5#	2-8	2-8	2-8#	2-10	2-10	2-10#	2-13	2-13	2-13#	2-16	
	2-16	2-16#	2-18	2-18	2-18#	2-21	2-22	2-22	2-22#	2-25	2-25	2-25#	2-27	2-27	
	2-27#	2-32	2-32	2-32#	2-32	2-32	2-32#	2-39	2-39	2-39#	2-41	2-41	2-41#	2-43	
	2-43	2-43#	2-45	2-45	2-45#	2-47	2-47	2-47#	2-49	2-49	2-49#	2-55	2-55	2-55#	
	2-56	2-56	2-56#	2-60	2-63	2-66	2-66	2-66#	2-68	2-68	2-68#	2-75	2-75	2-75#	

	2-78	2-78	2-78#	2-81	2-81	2-81#	2-83	2-83	2-83#	2-86	2-86	2-86#	2-87	2-87
	2-87#	2-91	2-91	2-91#	2-92	2-92	2-92#	2-94	2-94	2-94#	2-96	2-96	2-96#	2-99
	2-99	2-99#	2-101	2-101	2-101#	2-102	2-102	2-102#	2-106	2-106	2-106#	2-111	2-111	2-111#
	2-114	2-114	2-114#	2-117	2-117	2-117#	2-119	2-119	2-119#	2-124	2-124	2-124#	2-126	2-126
	2-126#	2-128	2-128	2-128#	2-130	2-130	2-130#	2-132	2-132	2-132#	2-134	2-134	2-134#	2-136
	2-136	2-136#	2-138	2-138	2-138#	2-140	2-140	2-140#	2-142	2-142	2-142#	2-144	2-144	2-144#
	2-146	2-146	2-146#	2-149	2-149	2-149#	2-152	2-152	2-152#	2-157	2-157	2-157#	2-160	2-160
	2-160#	2-162	2-162	2-162#	2-163	2-163	2-163#	2-166	2-166	2-166#	2-169	2-169	2-169#	2-171
	2-171	2-171#	2-173	2-173	2-173#	2-175	2-175	2-175#	2-176	2-176	2-176#	2-178	2-178	2-178#
	2-180	2-180	2-180#	2-182	2-182	2-182#	2-185	2-185	2-185#	2-188	2-188	2-188#	2-189	2-189
	2-189#	2-191	2-191	2-191#	2-193	2-193	2-193#	2-195	2-195	2-195#	2-198	2-198	2-198#	2-200
	2-200	2-200#	2-202	2-202	2-202#	2-204	2-204	2-204#	2-206	2-206	2-206#	2-208	2-208	2-208#
	2-211	2-211	2-211#	2-213	2-213	2-213#	2-215	2-215	2-215#	2-217	2-217	2-217#	2-219	2-219
	2-219#	2-221	2-221	2-221#	2-223	2-223	2-223#	2-225	2-225	2-225#	2-227	2-227	2-227#	2-229
	2-229	2-229#	2-233	2-233	2-233#	2-235	2-235	2-235#	2-237	2-237	2-237#	2-239	2-239	2-239#
	2-241	2-241	2-241#	2-244	2-244	2-244#	2-247	2-247	2-247#	2-250	2-250	2-250#	2-253	
ERREAD	2-176#	2-176#												
ERREN	2-49#	2-49#												
ERREP	2-91#	2-91#												
ERROM1	2-233#	2-233#												
ERROM2	2-235#	2-235#												
ERROR	12-5	12-11	12-12	12-24	12-31	12-32	12-39	12-40	12-46	12-47	13-23#			
ERRSKP	4-29	4-31#												
ERSEC	2-117#	2-117#												
ERSFE	2-83#	2-83#												
ERSHRP	2-25#	2-25#												
ERSOR	2-18#	2-18#												
ERSTGL	2-128#	2-128#												
ERSTOP	2-237#	2-237#												
ERSUBC	2-32#	2-32#												
ERSYNX	2-241#	2-241#												
ERTABO	2-217#	2-217#												
ERTERM	2-239#	2-239#												
ERTMOS	2-202#	2-202#												
ERTRNG	2-16#	2-16#												
ERUNOF	2-111#	2-111#												
ERUP	2-10#	2-10#												
ERUPN	2-75#	2-75#												
ERVAL	2-92#	2-92#												
ERWBYT	2-41#	2-41#												
ERWRT	2-162#	2-162#												
ERWSFL	2-119#	2-119#												
EXT	12-26	13-24#												
EXTND	12-20	13-25#												
FIELD	12-41	13-26#												
FILE	12-30	12-37	13-27#											
FIX1	3-7#													
FMT	12-36	12-40	12-41	13-28#										
FN	12-13	12-15	12-23	13-29#										
FND	12-30	12-34	13-30#											
FULL	12-22	13-31#												
GLBFLG	4-4#	4-11	4-13%											
GPBUS	12-39	13-32#												
HLDEOF	3-3#	7-38												
IMED	12-45	13-33#												
IN	12-41	12-42	12-45	13-34#										
INAGTE	3-4#	10-6												

CROSS REFERENCE TABLE (CREF V01-03)

```
INLIMD  7-8     7-13#
INLRCL  7-11#   7-14
INTRPT  12-23   13-35#
INVLD   12-10   12-14   12-15   12-16   12-17   12-29   12-36   12-37   12-38   13-36#
IO      12-38   12-39   13-37#
IS      12-21   12-22   12-23   12-33   13-38#
JMPAX   3-10#   4-64
KEYSTK  3-5#    4-32%   4-33%   4-34%   4-35%
L       13-10#  13-17   13-32   13-32   13-42   13-43   13-53   13-69
LS      5-41    13-9#   13-21   13-42
L.      5-44    13-9#   13-37
LA      13-9#   13-14   13-15   13-16   13-17   13-17   13-18   13-19   13-20   13-20   13-24   13-28   13-32   13-33
        13-36   13-42   13-43   13-43   13-49   13-51   13-54   13-58   13-61   13-63   13-68   13-68
LASTER  2-253#  4-19    4-21
LB      13-9#   13-14   13-21   13-32   13-44   13-68
LC      13-9#   13-17   13-18   13-19   13-21   13-29   13-32   13-57   13-69
LD      13-9#   13-14   13-18   13-19   13-20   13-21   13-25   13-25   13-26   13-30   13-33   13-35   13-36   13-54
        13-55   13-66   13-66   13-69
LDAX    3-10#   4-62
LDBX    3-10#   4-25
LDXX    3-10#
LE      13-9#   13-14   13-16   13-17   13-18   13-21   13-22   13-23   13-24   13-24   13-25   13-25   13-25   13-26
        13-27   13-32   13-32   13-33   13-33   13-35   13-35   13-39   13-41   13-42   13-42   13-43   13-44   13-48
        13-49   13-50   13-54   13-55   13-55   13-57   13-57   13-58   13-59   13-61   13-61   13-64   13-66   13-66
        13-68   13-69   13-69   13-69
LENMSG  12-49#
LENURD  13-71#
LF      13-9#   13-22   13-26   13-27   13-28   13-29   13-30   13-31   13-32   13-50   13-66
LG      13-9#   13-16   13-18   13-32   13-40   13-42   13-43   13-51
LH      13-9#   13-58
LI      13-9#   13-18   13-21   13-21   13-21   13-26   13-27   13-29   13-32   13-33   13-33   13-34   13-35   13-36
        13-36   13-37   13-38   13-39   13-49   13-52   13-53   13-55   13-59   13-66   13-67   13-68   13-69
LINE    12-10   12-18   12-29   12-42   12-43   12-45   13-39#
LJ      13-9#
LK      13-9#
LL      13-9#   13-17   13-17   13-24   13-26   13-27   13-31   13-31   13-36   13-39   13-40   13-50   13-68
LM      13-9#   13-16   13-17   13-19   13-19   13-21   13-28   13-33   13-33   13-41   13-41   13-42   13-43   13-44
        13-51   13-56   13-61   13-64
LN      13-9#   13-15   13-16   13-17   13-19   13-21   13-21   13-24   13-25   13-29   13-29   13-30   13-32   13-34
        13-35   13-36   13-39   13-40   13-44   13-45   13-46   13-47   13-48   13-49   13-52   13-61   13-63   13-66
        13-66   13-67
LO      13-9#   13-14   13-19   13-21   13-22   13-23   13-28   13-29   13-30   13-37   13-40   13-41   13-45   13-46
        13-47   13-48   13-49   13-49   13-50   13-50   13-51   13-53   13-53   13-56   13-62   13-65   13-65   13-69
LONG    12-28   13-40#
LP      13-9#   13-21   13-32   13-35   13-43   13-48   13-49   13-51   13-52   13-53   13-58   13-62   13-69
LQ      13-9#   13-55   13-60
LR      13-9#   13-14   13-16   13-18   13-18   13-21   13-23   13-23   13-23   13-24   13-28   13-32   13-35   13-35
        13-41   13-44   13-49   13-50   13-51   13-51   13-52   13-53   13-53   13-54   13-55   13-55   13-56   13-57
]]]]]
        13-60   13-68   13-69   13-69
LS      13-9#   13-21   13-21   13-21   13-32   13-38   13-42   13-42   13-57   13-58   13-59   13-60   13-61   13-62
        13-63   13-64   13-64
LT      13-9#   13-14   13-16   13-18   13-20   13-21   13-24   13-25   13-28   13-29   13-32   13-33   13-35   13-35
        13-43   13-46   13-49   13-52   13-53   13-57   13-61   13-61   13-61   13-62   13-63   13-64   13-65   13-67
        13-69   13-69   13-69
LU      13-9#   13-16   13-21   13-29   13-30   13-31   13-32   13-35   13-44   13-55   13-66   13-67
LV      13-9#   13-36   13-50   13-68
LW      13-9#   13-50   13-69
```

CROSS REFERENCE TABLE (CREF V01-03)

```
LX      13-9#   13-24   13-25   13-63
LY      13-9#   13-41   13-63   13-64
LZ      13-9#   13-59
M1      2-5     2-8     2-10    2-13    2-16    2-18    12-5#
M10     2-39    2-41    2-43    2-45    2-47    2-49    2-55    2-56    2-66    12-14#
M11     2-68    2-75    2-78    2-81    2-83    2-86    2-87    2-91    2-92    2-94    2-96    12-15#
M12     2-99    2-101   12-16#
M13     2-102   12-17#
M14     2-106   12-18#
M15     2-111   12-19#
M16     2-114   12-20#
M17     2-117   12-21#
M18     2-119   12-22#
M19     2-124   12-23#
M2      2-237   12-6#
M20     2-126   2-128   2-132   2-229   2-247   2-250   12-24#
M21     2-130   12-25#
M22     2-134   2-136   2-138   12-26#
M23     2-140   12-27#
M24     2-142   2-144   12-28#
M25     2-146   12-29#
M26     2-149   12-30#
M27     2-152   12-31#
M28     2-157   2-160   12-32#
M29     2-162   12-33#
M3      2-239   12-7#
M30     2-169   12-34#
M31     2-163   12-35#
M32     2-166   2-171   12-36#
M33     2-173   12-37#
M34     2-175   2-176   2-178   2-180   2-182   2-185   2-188   12-38#
M35     2-189   12-39#
M36     2-191   2-193   2-195   2-198   2-200   2-202   2-204   2-206   2-208   2-211   2-213   2-215   2-217   2-219
        2-221   2-223   2-225   12-40#
M37     2-227   12-41#
M4      2-241   12-8#
M5      2-244   12-9#
M6      2-22    12-10#
M7      2-25    12-11#
M8      2-27    2-32    12-12#
M9      2-37    12-13#
MEM     12-22   13-41#
MINED   7-13    12-45#
MINLN   7-10    12-42#
MMSGN   9-8     12-44#
MPRLN   7-29    12-43#
MROM1   2-233   12-47#
MROM2   2-235   12-48#
MSG     12-44   13-42#
MSGLP   4-50#   4-57    4-65
MSGORG  4-59    12-5    12-5#   12-6    12-7    12-8    12-9    12-10   12-11   12-12   12-13   12-14   12-15   12-16
        12-17   12-18   12-19   12-20   12-21   12-22   12-23   12-24   12-25   12-26   12-27   12-28   12-29   12-30
        12-31   12-32   12-33   12-34   12-35   12-36   12-37   12-38   12-39   12-40   12-41   12-42   12-43   12-44
        12-45   12-46   12-47   12-48   12-49
MSGTBL  2-5#    4-24
MSNTX   8-24    8-50    12-46#
MT      12-30   12-31   12-32   12-33   12-35   12-36   13-43#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

CMD 1-103# 11-13 11-14 11-15 11-16 11-17 11-18 11-19 11-20

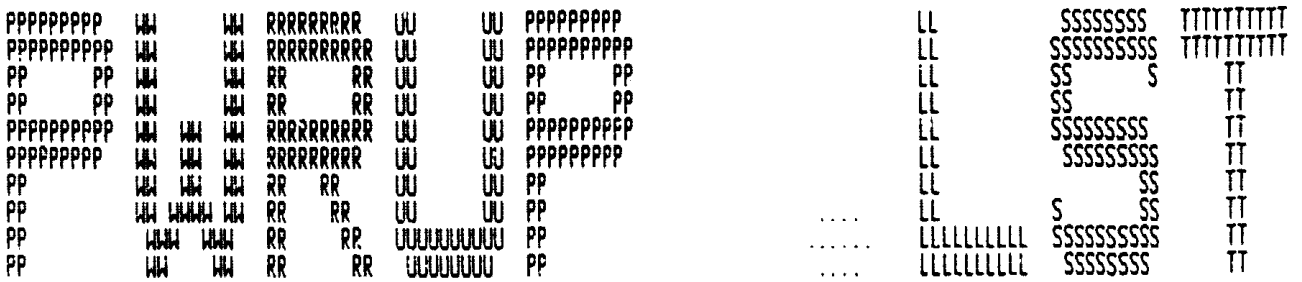
DERR 1-44# 2-60 2-63

ERR 1-35# 2-5 2-8 2-10 2-13 2-16 2-18 2-22 2-25 2-27 2-32 2-37 2-39 2-41
2-43 2-45 2-47 2-49 2-55 2-56 2-66 2-68 2-75 2-78 2-81 2-83 2-86 2-87
2-91 2-92 2-94 2-96 2-99 2-101 2-102 2-106 2-111 2-114 2-117 2-119 2-124 2-126
2-128 2-130 2-132 2-134 2-136 2-138 2-140 2-142 2-144 2-146 2-149 2-152 2-157 2-160
2-162 2-163 2-166 2-169 2-171 2-173 2-175 2-176 2-178 2-180 2-182 2-185 2-188 2-189
2-191 2-193 2-195 2-198 2-200 2-202 2-204 2-206 2-208 2-211 2-213 2-215 2-217 2-219
2-221 2-223 2-225 2-227 2-229 2-233 2-235 2-237 2-239 2-241 2-244 2-247 2-250

MSG 1-51# 12-5 12-6 12-7 12-8 12-9 12-10 12-11 12-12 12-13 12-14 12-15 12-16 12-17
12-18 12-19 12-20 12-21 12-22 12-23 12-24 12-25 12-26 12-27 12-28 12-29 12-30 12-31
12-32 12-33 12-34 12-35 12-36 12-37 12-38 12-39 12-40 12-41 12-42 12-43 12-44 12-45
12-46 12-47 12-48

SEI 1-3#

WRD 1-79# 13-14 13-15 13-16 13-17 13-18 13-19 13-20 13-21 13-22 13-23 13-24 13-25 13-26
13-27 13-28 13-29 13-30 13-31 13-32 13-33 13-34 13-35 13-36 13-37 13-38 13-39 13-40
13-41 13-42 13-43 13-44 13-45 13-46 13-47 13-48 13-49 13-50 13-51 13-52 13-53 13-54
13-55 13-56 13-57 13-58 13-59 13-60 13-61 13-62 13-63 13-64 13-65 13-66 13-67 13-68
13-69


PPPPPPPPP WW WW RRRRRRRR UU UU PPPPPPPPP LL SSSSSSS TTTTTTTTTT
PPPPPPPPPP WW WW RRRRRRRRRR UU UU PPPPPPPPPP LL SSSSSSSSS TTTTTTTTTT
PP PP WW WW RR RR UU UU PP PP LL SS S TT
PP PP WW WW RR RR UU UU PP PP LL SS TT
PPPPPPPPPP WW WW WW RRRRRRRRR UU UU PPPPPPPPPP LL SSSSSSSS TT
PPPPPPPPP WW WW WW RRRRRRRR UU UU PPPPPPPPP LL SSSSSSSS TT
PP WW WW WW RR RR UU UU PP LL SS TT
PP WW WWWW WW RR RR UU UU PP .... LL S SS TT
PP WWWW WWW RR RR UUUUUUUUU PP ..... LLLLLLLLL SSSSSSSS TT
PP WW WW RR RR UUUUUUU PP .... LLLLLLLLL SSSSSSS TT

14-OCT-76

```
                        .TITLE  PWRUP   POWER UP ROUTINE
                        .IDENT  /BE2058/
                        .GLOBL  PWRUP
                        ;
                        .MACRO  PIA     NM,DDR,CTRLR,SRTN,ADDR
                        .GLOBL  PIA'NM
                PIA'NM  =       HO'ADDR
                        .IF     NB,DDR
                        .WORD   PIA'NM,SRTN
                        .BYTE    HO'DDR, HO'CTRLR
                        .ENDC
                        .ENDM
                        ;
                        .GLOBL  IDLE
                        .GLOBL  HEP             ;ADDR OF PTR TO END OF RAM
                        .GLOBL  STPTR           ;SYMBOL TABLE PTR
                        .GLOBL  LSP             ;LOW STACK POINTER
                        .GLOBL  SBP             ;BASE OF MAIN STACK
                        .GLOBL  ISB             ;IMAGE STRING BASE
                        .GLOBL  EOSTG           ;STACK TAG FOR END OF STACK
                        .GLOBL  CTKN            ;CURRENT EVAL TOKEN
                        .GLOBL  PGMEND          ;END OF SYSTEM RAM
                        .GLOBL  XEQSP           ;EXECUTIN STACK POINTER
                        .GLOBL  XEQSTK          ;DATA SPACE
                        .GLOBL  EDTEND,EDTBFR
                        .GLOBL  R10,R11,R12,R13,R14,R20
                        .GLOBL  JMPX,LDAX,LDBX,LDXX,STAX,JMPAX
                        .GLOBL  96X,A11X,D5X
                        .GLOBL  DREXTA,DREXTB
                        .GLOBL  BANKSW
                        .GLOBL  BANK
                        .GLOBL  PIATBL
                        .GLOBL  INITX           ;SYSTEM DEFAULT SET UP ROUTINE
                        .GLOBL  BNKADR          ;BASE ADDR FOR BANK SW AREA
                        .GLOBL  KEYQUE          ; KEYBOARD QUE
                        .GLOBL  QIN             ; POINTERS
                        .GLOBL  KBINT           ;KEYBOARD SERVICE ROUTINE
                        .GLOBL  CNTL            ;KEY BOARD CONTROLS
                        .GLOBL  LINELN          ;LINE WIDTH ON CRT
                        .GLOBL  SYSERR          ;SYSTEM ERROR ROUTINE
                        .GLOBL  CRASH           ;LATER HOLDING AREA FOR SYSERR DATA
                        .GLOBL  PSHFPN,PULFPN
                        .GLOBL  ZX              ;TWOBYTES OF ZEROS IN PAGE ZERO
                        .GLOBL  ERRCD           ;SYSTEM ERROR CODE BYTE
                        .GLOBL  SRQRDY,IECE01
                        .GLOBL  HIRAMZ          ;HI RAM ADDRESS DEFINATION
                        .GLOBL  ACIA            ;COM PACK ACIA ADDR
                        ;
        0020    HIRAM   =       32.             ;LAST ADDR IN RAM IN 1 K UNITS
        7FE5    HIRAMZ  =       HIRAM*1024.-11. ;LAST ADDR I CAN USE IN BYTES
        000F    MASK    =        HOF            ;MASK FOR ZERO BITS IN VALID MEMORY
                        ;                        ADDRESS.  MLT OF 4K BOUNDRIES.
```

```
                                ;
                                ;       INITBL FORMAT
                                ;       2 BYTES - ADDR IN RAM THIS DATA GOES AT
                                ;       1 BYTE - LENGTH OF DATA IN BYTES
                                ;       N BYTES - DATA TO GO TO RAM
                                ;       NOTE.  IF RAM ADDR HAS BIT 7 SET THIS IS END OF TABLE FLAG
                                ;
                                ;       PIATAB FORMAT - TABLE IN POWER UP
                                ;       2 BYTES - PIA ADDRESS
                                ;       2 BYTES - INTERRUPT SERVICE ROUTINE ADDRESS
                                ;       1 BYTE - DATA DIRECTION REGISTER DATA
                                ;       1 BYTE - CONTROL REGISTER DATA
                                ;
                                ;
                                ;       SWIERR IS THE ENTRY POINT THAT SWI INTERRUPTS PASS CONTROL TO.
                                ;
                                ;       SYSERR IS THE LOCATION THE SYSTEM PASSES CONTROL TO WHEN IT
                                ;       IS LOST.
                                ;
                                ;               'NO PROBLEM IS TOO BIG OR TOO COMPLICATED TO BE RUN AWAY
                                ;                       FROM.'
                                ;
                                ;       PWRUP IS THE ENTRY POINT FOR POWER UP RESTART.  IT SETS
                                ;       UP THE BASIC SYSTEM CONTROL VARIABLES AND THEN PASSES CONTROL TO
                                ;       THE IDLE LOOP.
                                ;
0000  8D  00          SWIERR:   BSR     SYSERR          ;MARK SWI IF IT HAPPENS
0002                  SYSERR:   SEI                     ;SYSTEM ERROR ROUTINE
0002  01  0F          .BYTE     01,17
0004  9F  00                    STS     0,D             ;SAVE STACK POINTER
0006  DF  02                    STX     2,D             ;INDEX REG
0008  30                        TSX
0009  EE  00                    LDX     0,X
0008  DF  04                    STX     4,D             ;PROGRAM COUNTER
000D  97  06                    STA A   6,D             ;ACC-A
000F  D7  07                    STA B   7,D             ;ACC-B
0011  96  00G                   LDA A   BANK,D          ;CURRENT BANK
0013  8A  80                    ORA A   128.,I          ;MARK AS SYSERR
0015  97  08                    STA A   8.,D            ;CONDITION CODE REG (CAN'T BE ALL ZEROS)
0017  96  00G                   LDA A   CTKN,D
0019  97  09                    STA A   9.,D            ;CURRENT EVAL TOKEN
001B  20  03                    BRA     FAKEIT          ;NOW DO A FAKE POWERUP SEQUINCE
                                ;
0010  7F  0008        PWRUP:    CLR     8.              ;SO THIS IS POWER UP AND NOT SYSERR
                                ;     XXX
                                ;    XXX
0020  8E  0000G       FAKEIT:   LDS     R20,I           ;XXXXXXXXXXX LOAD STACK POINTER XXXXXXXXXX
                                ;    XXX
                                ;     XXX
0023  7F  87C1                  CLR     BANKSW+1        ;SET BANK SWITCH TO MAIN ROM
0026  86  38                    LDA A   56.,I           ;BANK SWITCH MAY BE A PIA
0028  B7  87C0                  STA A   BANKSW
002B  86  04                    LDA A   4,I
002D  B7  87C1                  STA A   BANKSW+1
0030  7F  87C0                  CLR     BANKSW
```

```
                                ;
                                ;RESET ACIA IN COM PACK
0033  86 03          LDA A  3,I
0035  B7 8206        STA A  ACIA
                                ;CLEAR RAM
0038  CE 7FF5        HIRAM2: LDX    HIRAM,I
0038  6F 0A          PRLPA:  CLR    10.,X
003D  09                      DEX
003E  26 FB                   BNE    PRLPA
                                ;
                                ;SEARCH FOR END OF INSTALLED RAM
0040  FE 0039'                LDX    HIRAM2+1
0043  86 FF                   LDA A  255.,I
                                ;SET UP FOR LATER
0045  6D 0A          PRLPB:  TST    10.,X
0047  26 06                   BNE    PRSKPB
0049  A7 0A                   STA A  10.,X
                                ;SEE IF I CAN SET LOCATIONS
004B  A1 0A                   CMP A  10.,X
004D  27 05                   BEQ    PRHLT
004F  09              PRSKPB: DEX
0050  26 F3                   BNE    PRLPB
                                ;
                                ;I AM IN REAL TROUBLE NOW
0052  20 FE           PRFAIL: BRA    PRFAIL
                                ;
                                ;LAST ADDR+1 FOR EASY CODEING
0054  BD 0000G        PRHLT:  JSR    A11X
0057  DF 00G                  STX    HEP,D
                                ;SAVE FOR SYSTEM USE
0059  96 01G                  LDA A  HEP+1,D
                                ;SECOND BYTE MUST BE ZERO
005B  26 F5                   BNE    PRFAIL
005D  96 00G                  LDA A  HEP,D
                                ;SEE IF IT IS 4K BOUNDRY
005F  85 0F                   BIT A  MASK,I
0061  26 EF                   BNE    PRFAIL
                                ;
                                ;ALLOCATE SYMBOL TABLE SPACE
0063  BD 0000G                JSR    DEX
0066  DF 00G                  STX    STPTR,D
                                ;SET SYMBOL TABLE PTR
0068  09                      DEX
0069  DF 00G                  STX    SBP,D
                                ;SET STACK BASE PTR
006B  9E 00G                  LDS    SBP,D
                                ;SET STACK PTR
006D  86 00G                  LDA A  EOSTG,I
                                ;TAG END OF STACK
006F  36                      PSH A
                                ;
                                ;SET UP NONE ZERO RAM LOCATIONS
0070  CE 01A1'                LDX    INTTBL,I
0073  DF 00G                  STX    R10,D
0075  DE 00G          PRLPJ:  LDX    R10,D
                                ;INTTBL ENTRY PTR
0077  EE 00                   LDX    0,X
                                ;DATA ADDR - TARGET AREA
0079  2B 1E                   BMI    PRSKPJ
                                ;IF BIT 7 SET THIS IS END OF TABLE
007B  DF 00G                  STX    R11,D
                                ;SAVE FOR LATER
007D  DE 00G                  LDX    R10,D
007F  A6 02                   LDA A  2,X
                                ;BYTE COUNT TO MOVE
0081  08                      INX
                                ;ADDR AND COUNT IS USED
0082  08                      INX
0083  08                      INX
0084  DF 00G                  STX    R10,D
                                ;
0086  DE 00G          PRLPK:  LDX    R10,D
                                ;INPUT DATA ADDR
0088  E6 00                   LDA B  0,X
                                ;DATA
008A  08                      INX
                                ;UPDATE PTR
008B  DF 00G                  STX    R10,D
008D  DE 00G                  LDX    R11,D
                                ;TARGET ADDR
008F  E7 00                   STA B  0,X
0091  08                      INX
0092  DF 00G                  STX    R11,D
0094  4A                      DEC A
                                ;COUNT BYTES
```

```
0095  26  EF             BNE    PRLPK        ;MOVE DATA IN ENTRY
0097  20  DC             BRA    PRLPJ        ;TRY NEXT ENTRY
                ;
0099  DE  00G    PRSKPJ: LDX    ZX,D         ;MOVE SYSERR DATA
009B  BD  0000G          JSR    PSHFPN
009E  CE  0000G          LDX    CRASH,I
00A1  BD  0000G          JSR    PULFPN
00A4  DE  08             LDX    8.,D         ;LAST TWO BYTES
00A6  FF  0008G          STX    CRASH+8.
                         .GLOBL PUPTAP
00A9  BD  0000G          JSR    PUPTAP
00A   DE  00G    PIALP:  LDX    R12,D        ;REGISTERS WERE SET UP BY INTTBL
00AE  A6  00             LDA A  0,X          ;GET PIA ADDR
00B0  E6  01             LDA B  1,X
00B2  DE  00G            LDX    R13,D
00B4  A7  00             STA A  0,X
00B6  E7  01             STA B  1,X
00B8  DE  00G            LDX    R12,D
00BA  A6  02             LDA A  2,X          ;GET SERVICE ROUTINE ADDR
00BC  E6  03             LDA B  3,X
00BE  DE  00G            LDX    R13,D
00C0  A7  03             STA A  3,X
00C2  E7  04             STA B  4,X
00C4  DE  00G            LDX    R12,D
00C6  A6  04             LDA A  4,X          ;GET DDR CONTROL BITS
00C8  E6  05             LDA B  5,X          ;GET CONTROL REG BITS
00CA  EE  00             LDX    0,X          ;GET PIA ADDR
00CC  36                 PSH A
00CD  86  04             LDA A  4,I          ;SET CONTROL TO DATA REG
00CF  A7  01             STA A  1,X
00D1  86  FF             LDA A  255.,I       ;SET ALL LINES HIGH
00D3  A7  00             STA A  0,X
00D5  32                 PUL A               ;AND RESTORE GRABAGE IN ACCA
00D6  6F  01             CLR    1,X          ;SET CONTROL REG TO ZERO
00D8  A7  00             STA A  0,X          ;SET DDR
00DA  E7  01             STA B  1,X          ;SET CONTROL REG
00DC  DE  00G            LDX    R12,D        ;UPDATE PTRS
00DE  BD  0000G          JSR    A6X
00E1  DF  00G            STX    R12,D
00E3  DE  00G            LDX    R13,D
00E5  BD  0000G          JSR    A6X
00E8  DF  00G            STX    R13,D
00EA  7A  0000G          DEC    R14          ;LOOP COUNTER
00ED  26  BD             BNE    PIALP
00EF  7F  87AA           CLR    PIALT
00F2  BD  0000G          JSR    INITMT
                         .GLOBL INITMT
00F5  BD  0000G          JSR    REWIND
                         .GLOBL REWIND
00F8  7F  0000G          CLR    ERRCD
                         .GLOBL CRTRST
00FB  BD  0000G          JSR    CRTRST
00FE  BD  0000G          JSR    DSHOME
                         .GLOBL DSHOME
0101  86  08             LDA A  8.,I         ; SET UP IEC BUS
0103  B7  8782           STA A  PIASRO
0106  CE  00023          LDX    BNKADR+2,I   ;POWER UP ROM PACK CALL ADDR
```

```
  171  0109  BD  0000G          JSR   INITX     ;SET UP REST OF SYSTEM
  172  010C  B6  0008G          LDA A CRASH+8.  ;IS IT SYSERR
  173  010F  26  07             BNE   UGH       ;IT IS
  174  0111  B6  87A8     SYSGO: LDA A PIAKB     ; READ THE KEYBOARD AND LEAVE THIS HERE!!
  175  0114  0E                  CLI            ;ENABLE INTERRUPTS
  176  0115  7E  0000G          JMP   IDLE      ;GO TO CURSOR LOOP
```

```
                                        ;
                                        ;       DUMP REG ROUTINE FOR SYSTEM ERRORS
                                        ;
                                        ;       .GLOBL  CTLCHR,TMP1,TMPHY,TMPHX,VECTOR,DSPCPY
                                        ;
                0020                    SPACE   =       H20             ;SPACE CHARACTER
                                        ;
0118    7C      0000G           UGH:    INC     TMP1            ;SET MODE
0118    DE      00G                     LDX     ZX.D            ;MOVE TO BOTTOM OF SCREEN
011D    DF      00G                     STX     TMPHY,D
011F    BD      0000G                   JSR     VECTOR          ;MOVE CURSOR
0122    CE      017E'                   LDX     UGHMSG,I        ;SAY HI
0125    DF      00G             UGHLP:  STX     R10,D
0127    A6      00                      LDA A   0,X             ;GET AND TEST CHAR
0129    27      08                      BEQ     UGHSKP          ;NO MORE
012B    BD      0000G                   JSR     CTLCHR
012E    DE      00G                     LDX     R10,D
0130    08                              INX
0131    20      F2                      BRA     UGHLP
                                        ;
0133    CE      0000G           UGHSKP: LDX     CRASH,I         ;STUFF IS IN LOW MEMORY
0136    DF      00G                     STX     R10,D
0138    8D      14                      BSR     HEXTWO          ;SP
013A    8D      12                      BSR     HEXTWO          ;X-REG
013C    8D      10                      BSR     HEXTWO          ;PC
013E    8D      17                      BSR     HEXONE          ;A-REG
0140    8D      15                      BSR     HEXONE          ;B-REG
0142    8D      13                      BSR     HEXONE          ;CC-REG
0144    8D      11                      BSR     HEXONE          ;CTKN
0146    BD      0000G                   JSR     DSPCPY          ;MAKE HARD COPY
0149    BD      0000G                   JSR     DSHOME          ;RESET CURSOR
014C    20      C3                      BRA     SYSGO
                                        ;
014E    DE      00G             HEXTWO: LDX     R10,D           ;OUTPUT TWO BYTES
0150    A6      00                      LDA A   0,X
0152    08                              INX                     ;BUMP POINTER FOR NEXT TIME
0153    DF      00G                     STX     R10,D
0155    8D      0E                      BSR     HEXOUT
0157    DE      00G             HEXONE: LDX     R10,D           ;OUTPUT ONE BYTE
0159    A6      00                      LDA A   0,X
015B    08                              INX
015C    DF      00G                     STX     R10,D
015E    8D      05                      BSR     HEXOUT
0160    86      20                      LDA A   SPACE,I         ;TRAILING BLANK
0162    7E      0000G                   JMP     CTLCHR
                                        ;
0165    97      00G             HEXOUT: STA A   R11,D           ;SAVE BYTE
0167    8D      04                      BSR     HEXL
0169    96      00G                     LDA A   R11,D
016B    20      04                      BRA     HEXR
                                        ;
016D    44                      HEXL:   LSR A                   ;SHIFT HIGH ORDER TO LOW ORDER
016E    44                              LSR A
016F    44                              LSR A
0170    44                              LSR A
0171    84      0F              HEXR:   AND A   15,I            ;ISOLATE FOUR BITS
0173    8B      30                      ADD A   '0,I            ;BIAS TO DIGITS
```

```
58          0175  81  39                CMP A   '9,I            ;SHOULD IT BE LETTER
59          0177  23  02                BLS     HEXOK           ;NO
60          0179  8B  07                ADD A   'A-'0-10.,I     ;FIX IT UP
61          017B  7E  00000G      HEXOK: JMP     CTLCHR          ;TO OVST AND RETURN
62                                      ;
63          017E  53  59  53  UGHMSG: .ASCIZ  /SYSTEM ERROR - MESSAGE NUMBER 0 - /
      0181  54  45  4D
      0184  20  45  52
      0187  52  4F  52
      018A  20  2D  20
      018D  4D  45  53
      0190  53  41  47
      0193  45  20  4E
      0196  55  4D  42
      0199  45  52  20
      019C  30  20  20
      019F  20  00
```

```
                                INTTBL: .WORD   JMPX            ;TARGET ADDR
01A3    24                              .BYTE   36.             ;DATA LENGTH
                                ;
01A4    B7  0004G                       STA A   JMPX+4          ;DATA
01A7    EE  00                          LDX     0,X
01A9    6E  00                          JMP     0,X
                                ;
01AB    B7  0004G                       STA B   LDXX+4
01AE    EE  00                          LDX     0,X
01B0    39                              RTS
                                ;
01B1    B7  0004G                       STA A   LDAX+4
01B4    A6  00                          LDA A   0,X
01B6    39                              RTS

01B7    B7  0004G                       STA A   LDBX+4
01BA    E6  00                          LDA B   0,X
01BC    39                              RTS
                                ;
01BD    B7  0004G                       STA B   STAX+4
01C0    E7  00                          STA B   0,X
01C2    39                              RTS
                                ;
01C3    B7  0004G                       STA A   JMPAX+4
01C6    6E  00                          JMP     0,X
                                ;
01C8    0000G                           .WORD   DREXTA          ;FANCY JUMPS FOR EXITS
01CA    01  7E                          .BYTE   1, H7E
                                ;
01CC    0000G                           .WORD   DREXTB
01CE    01  7E                          .BYTE   1, H7E
                                ;
01D0    0000G                           .WORD   LSP             ;END OF SYSTEM RAM POINTER
01D2    04                              .BYTE   4
01D3    0000G                           WORD    PGNEND
01D5    0000G                           .WORD   PGNEND
                                ;
01D7    0000G                           WORD    EDTEND          ;EDITOR CONTROLS
01D9    02                              .BYTE   2
01DA    0048G                           .WORD   EDTBFR+72.
                                ;
01DC    0000G                           .WORD   R12             ;CONTROLS FOR PIA INIT LP
01DE    05                              .BYTE   5
01DF    0203'   0000G                   WORD    PIATAB,PIATBL
01E3    06                              .BYTE   NPIAS
                                ;
01E4    0000G                           WORD    QIN     ; INITIALIZE TYPE AHEAD Q POINTERS
01E6    04                              .BYTE   4
01E7    0003G                           .WORD   KEYQUE  ; QIN
01E9    0000G                           .WORD   KEYQUE  ; QOUT
                                ;
01E8    0000G                           .WORD   XEQSP
01ED    02                              .BYTE   2
01EE    0002G                           .WORD   XEQSTK+2
                                ;
01F0    0000G                           WORD    XEQSTK
01F2    02                              .BYTE   2
```

```
                                                .WORD   SYSERR
                                ;
                                                .WORD   CNTL
                                                .BYTE   1,255.
                                ;
                                                .WORD   ISB
                                                .BYTE   2,255.,255
                                ;
                                                .WORD   LINELN
                                                .BYTE   1,240.
                                ;
                                                .BYTE   128.            ;END OF TABLE
```

```
                0203                    PIATAB: PIA     HY,,,,878C
                0203                            PIA     LY,,,,878E
                0203                            PIA     HX,F3,2C,SYSERR,8794
     0203 8794  0002'                   .WORD   PIAHX,SYSERR
     0207   F3    2C                    .BYTE    HOF3, HO2C
                0209                            PIA     LX,FF,2C,SYSERR,8796
     0209 8796  0002'                   .WORD   PIALX,SYSERR
     020D   FF    2C                    .BYTE    HOFF, HO2C
                020F                            PIA     EOI,00,37,IECEOI,8780
     020F 8780  0000G                   .WORD   PIAEOI,IECEOI
     0213   00    37                    .BYTE    H000, H037
                0215                            PIA     SRQ,9F,37,SRQRDY,8782
     0215 8782  000'                    .WORD   PIASRQ,SRQRDY
     0219   9F    37                    .BYTE    H09F, H037
                0218                            PIA     MTA,,,,8798
                0218                            PIA     MTB,,,,879A
                0218                            PIA     KB,00,27,KBINT,87A8
     0218 87A8  0000G                   .WORD   PIAKB,KBINT
     021F   00    27                    .BYTE    H000, H027
                0221                            PIA     LT,E4,2C,SYSERR,87AA
     0221 87AA  0002'                   .WORD   PIALT,SYSERR
     0225   E4    2C                    .BYTE    HOE4, HO2C
                0006                    NPIAS   =       .-PIATAB/6
                                        ;
                87C0                    BANKSW  =       H87C0
                87C6                    ACIA    =       H87C6
                                        ;
                0000                            .ASECT
                                        ;
                FFF8                    TRAPS   =       HOFFF8          ; ORIGIN OF INTERRUPT VECTORS
                                        ;
                                                GLOBL   TRAPS,INTSRV,SWIERR,NMISRV,PWRUP
                                        ;
                FFF8                            . = TRAPS+0
     FFF8 0000G                                 WORD    INTSR           ; NORMAL I/O INTERRUPTS
                FFFA                            . = TRAPS+2
     FFFA 0000'                                 WORD    SWIERR          ; SOFTWARE INTERRUPTS ARE NOT USED
                FFFC                            . = TRAPS+4
     FFFC 0000G                                 WORD    NMISRV          ; MAG TAPE INTERRUPTS
                FFFE                            . = TRAPS+6
     FFFE 001D'                                 WORD    PWRUP           ; POWER UP RESTART
                                        ;
                001D'                           .END    PWRUP
```

```
PWRUP  POWER UP ROUTINE        RT-11 MMAC  VM02-10   14-OCT-76 01:40:15 PAGE 5+
SYMBOL TABLE

ACIA =   87C6 G      A11X = xxxxxx G      A6X  = xxxxxx G      BANK  = xxxxxx G      BANKSW=   87C0 G
BNKADR= xxxxxx G     CNTL = xxxxxx G      CRASH= xxxxxx G      CRTRST= xxxxxx G      CTKN  = xxxxxx G
CTLCHR= xxxxxx G     DREXTA= xxxxxx G     DREXTB= xxxxxx G     DSHOME= xxxxxx G      DSPCPY= xxxxxx G
D5X  = xxxxxx G      EDTBFR= xxxxxx G     EDTEND= xxxxxx G     EOSTG = xxxxxx G      ERRCD = xxxxxx G
FAKEIT   0020R       HEP  = xxxxxx G      HEXL     016DR       HEXOK    0178R        HEXONE    0157R
HEXOUT   0165R       HEXR     0171R       HEXTWO   014ER       HIRAM =   7FF5        HIRAMZ    0038RG
IDLE = xxxxxx G      IECEOI= xxxxxx G     INITMT= xxxxxx G     INITX = xxxxxx G      INTSRV= xxxxxx G
INTTBL   0181R       ISB  = xxxxxx G      JMPAX = xxxxxx G     JMPX  = xxxxxx G      KBINT = xxxxxx G
KEYQUE= xxxxxx G     LDAX = xxxxxx G      LDBX = xxxxxx G      LDXX  = xxxxxx G      LINELN= xxxxxx G
LSP  = xxxxxx G      MASK =   000F        NMISRV= xxxxxx G     NPIAS =   0006        PGNEND= xxxxxx G
PIAEOI=  8780 G      PIAHX =  8794 G      PIAHY =  878C G      PIAKB =  8788 G       PIALP     00ACR
PIALT =  879A G      PIALX =  8796 G      PIALY =  878E G      PIAMTA=  8798 G       PIAMTB=   879A G
PIASRQ=  8782 G      PIATAB   0203R       PIATBL= xxxxxx G     PRFAIL    0052R       PRHIT     0054R
PRLPA    0038R       PRLPB    0045R       PRLPJ    0075R       PRLPK     0086R       PRSKPB    00AFR
PRSKPJ   0099R       PSHFPN= xxxxxx G     PULFPN= xxxxxx G     PUPTAP= xxxxxx G      PWRUP     001DRG
QIN  = xxxxxx G      REWIND= xxxxxx G     R10  = xxxxxx G      R11  = xxxxxx G       R12  = xxxxxx G
R13  = xxxxxx G      R14  = xxxxxx G      R20  = xxxxxx G      SBP  = xxxxxx G       SPACE =   0020
SRQRDY= xxxxxx G     STAX = xxxxxx G      STPTR = xxxxxx G     SWIERR    0000RG      SYSERR    0002RG
SYSGO    0111R       TMPHX = xxxxxx G     TMPHY = xxxxxx G     TMP1 = xxxxxx G       TRAPS =   FFF8 G
UGH      0118R       UGHLP    0125R       UGHMSG   017ER       UGHSKP    0133R       VECTOR= xxxxxx G
XEQSP = xxxxxx G     XEQSTK= xxxxxx G     ZX   = xxxxxx G
. ABS.   FFFE     00
         0227     01

ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2859. WORDS
.SY:PWRUP/C<DK1:SELCL>,PWRUP
```

CROSS REFERENCE TABLE (CREF V01-03)

	5-11	5-22#	5-24#	5-26#	5-28#
A11X	1-43#	2-76			
A6X	1-43#	2-151	2-154		
ACLO	1-62#	2-58*	5-14#		
BANK	1-46#	2-38			
BANKSW	1-45#	2-51*	2-53*	2-55*	2-56* 5-13#
BNKADR	1-49#	2-120			
CNTL	1-53#	4-60			
CRASH	1-56#	2-119	2-122*	2-172	3-21
CRTRST	2-164#	2-165			
CTKN	1-36#	2-41			
CTLCHR	3-4#	3-16	3-45	3-61	
DSX	1-43#	2-84			
DREXTA	1-44#	4-27			
DREXTB	1-44#	4-30			
DSHOME	2-166	2-167#	3-31		
DSPCPY	3-4#	3-30			
EDTBFR	1-40#	4-40			
EDTEND	1-40#	4-38			
EOSTG	1-35#	2-89			
ERRCD	1-59#	2-163*			
FAKEIT	2-43	2-48#			
HEP	1-30#	2-77*	2-78	2-80	
HEXL	3-48	3-52#			
HEXOK	3-59	3-61#			
HEXONE	3-26	3-27	3-28	3-29	3-39#
HEXOUT	3-38	3-43	3-47#		
HEXR	3-50	3-56#			
HEXTWO	3-23	3-24	3-25	3-34#	
HIRAM	1-64#	1-65	1-65#	2-59	
HIRAM2	1-61#	2-59#	2-64		
IDLE	1-29#	2-176			
IECEOI	1-60#	5-5			
INITMT	2-159	2-160#			
INITX	1-48#	2-171			
INTSRV	5-20#	5-23			
INTTBL	2-92	4-1#			
ISB	1-34#	4-63			
JMPAX	1-42#	4-24*			
JMPX	1-42#	4-1	4-4*		
KBINT	1-52#	5-9			
KEYQUE	1-50#	4-49	4-50		
LDAX	1-42#	4-12*			
LDBX	1-42#	4-16*			
LDXX	1-42#	4-8*			
LINELN	1-54#	4-66			
LSP	1-32#	4-33			
MASK	1-66#	2-81			
NMISRV	5-20#	5-27			
NPTAS	4-45	5-11#			
PGMEND	1-37#	4-35	4-36		
PIAEOI	5-5	5-5#	5-5#		
PIAHX	5-3	5-3#	5-3#		
PIAHY	5-1#	5-1#			
PIAKB	2-174	5-9	5-9#	5-9#	
PIALP	2-125#	2-157			

CROSS REFERENCE TABLE (CREF V01-03)

PIALT 2-158# 5-10 5-10# 5-10#
PIALX 5-4 5-4# 5-4#
PIALY 5-2# 5-2#
PIGNTA 5-7# 5-7#
PIAMTB 5-8# 5-8#
PIASRQ 2-169# 5-6 5-6# 5-6#
PIATAB 4-44 5-1# 5-11
PIATBL 1-47# 4-44
PRFAIL 2-74 2-74# 2-79 2-82
PRHLT 2-70 2-76#
PRLPA 2-60# 2-62
PPLPB 2-66# 2-72
PRLPJ 2-94# 2-115
PRLPK 2-105# 2-114
PRSKPB 2-67 2-71#
PRSKPJ 2-96 2-117#
PSHFPN 1-57# 2-118
PULFPN 1-57# 2-120
PUPTAB 2-123# 2-124
PWRUP 1-18# 2-45# 5-20# 5-29 5-31
QIN 1-51# 4-47
R10 1-41# 2-93# 2-94 2-98 2-103# 2-105 2-108# 3-13# 3-17 3-22# 3-34 3-37# 3-39 3-42#
R11 1-41# 2-97# 2-109 2-112# 3-47# 3-49
R12 1-41# 2-125 2-131 2-137 2-150 2-152# 4-42
R13 1-41# 2-128 2-134 2-143 2-155#
R14 1-41# 2-156#
R20 1-41# 2-48
REWIND 2-161 2-162#
SBP 1-33# 2-87# 2-88
SPACE 3-6# 3-44
SRQRDY 1-60# 5-6
STAX 1-42# 4-20#
STPTR 1-31# 2-85#
SWIERR 2-29# 5-20# 5-25
SYSERR 1-55# 2-29 2-30# 4-58 5-3 5-4 5-10
SYSGO 2-174# 3-32
TMP1 3-4# 3-8#
TMPHX 3-4#
TMPHY 3-4# 3-10#
TRAPS 5-18# 5-20# 5-22 5-24 5-26 5-28
UGH 2-173 3-8#
UGHLP 3-13# 3-19
UGHMSG 3-12 3-63#
UGHSKP 3-15 3-21#
VECTOR 3-4# 3-11
XEQSP 1-38# 4-52
XEQSTK 1-39# 4-54 4-56
ZX 1-58# 2-117 3-9

CROSS REFERENCE TABLE (CREF V01-03)

P1Q 1-20# 5-1 5-2 5-3 5-4 5-5 5-6 5-7 5-8 5-9 5-10
SE1 1-3# 2-30

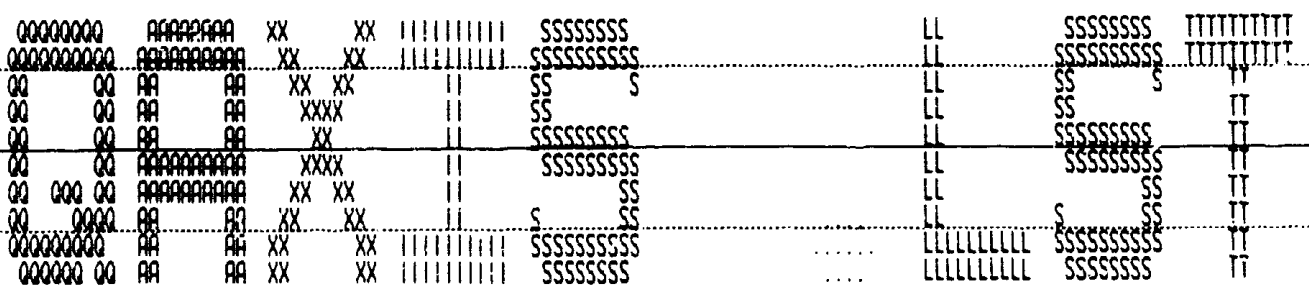


TABLE OF CONTENTS

```
                        .TITLE  QAXIS
                        .IDENT  /JMM025/
                        .GLOBL  QAXIS
                        .GLOBL  PLOTVL
                        .GLOBL  UNADR, ZORAW, ADRDEV, BERALC, OPRADR
                        .GLOBL  DOFP,FRET,FHEX,FHIN,FLEX,FLIN,FART,FSAV,FRES
0016                            FOUP= H16
0080                            FA  =200
00A0                            FS  =240
00C0                            FM  =300
00E0                            FD  =340
                        .GLOBL  XMINW   ;WINDOW
                        .GLOBL  XMAXW   ; "
                        .GLOBL  YMINW   ; "
                        .GLOBL  YMAXW   ; "
                        .GLOBL  XMINS   ;VIEWPORT IN GDU'S
                        .GLOBL  YMINS   ; "
                        .GLOBL  TEMPX
                        .GLOBL  TEMPY
                        .GLOBL  ATSNTG
                        .GLOBL  HALTR   ;CLEAN THE STACK ROUTINE
                        ;
                        .GLOBL  R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22
                        .GLOBL R22
                        .GLOBL  A9X
                        ;
                        .GLOBL  XLAST,YLAST
                        .GLOBL  XNEW,YNEW,FPZERO,EOLTG,CLRARG,SETARG,TYPARG
                        .GLOBL  PSHFPN,PULFPN
                        .GLOBL  PSHRET,RTRN
                        .GLOBL  FPCMP
                        .GLOBL  ERRCD
                        .GLOBL  ERDOMN
                        .GLOBL  A.SEC,A.STAT
                        .GLOBL  GRFDR
                        .GLOBL  SCRTCH
                        ;
                        ;
                        ;FLAG   01=Y TIC FLAG
                        ;       10=X TIC FLAG
                        ;       80=LEFT OR DOWN DRAW FLAG
                        ;
                        ;REGISTER USAGE --- POINTERS TO
                        ;       R 12    X INTERCEPT
                        ;       R 13    Y INTERCEPT
                        ;       R 14    X TIC INTERVAL
                        ;       R 15    Y TIC INTERVAL
                        ;
                        ;       R 16    Y INTERCEPT + DELTA
                        ;       R 17    Y INTERCEPT - DELTA
                        ;       R 18    X     "     +   "
                        ;       R 19    X     "     -   "
                        ;       R22     FLAG BYTES
                        ;
```

```
    1  0000  BD  0000G     QAXIS:  JSR     PSHRET
    2  0003  CE  0000G             LDX     SCRTCH,I        ;TEMP AREA
    3  0006  DF    00G             STX     R16,D
    4  0008  CE  0008G             LDX     SCRTCH+8.,I
    5  000B  DF    00G             STX     R17,D
    6  000D  CE  0010G             LDX     SCRTCH+16.,I
    7  0010  DF    00G             STX     R18,D
    8  0012  CE  0018G             LDX     SCRTCH+24.,I
    9  0015  DF    00G             STX     R19,D
   10  0017  CE  0000G             LDX     ZORAW,I
   11  001A  DF    00G             STX     OPRADR,D        ;DEFAULT PRI AND SEC ADDR
   12  001C  BD  0000G             JSR     ADRDEV          ;SET UP ADDR BUS
   13  001F  BD  0000G             JSR     BFRALC          ;BUFFER ALLOCATE FOR I/O SYSTEM
   14  0022  7F  00G0G             CLR     R22             ;CLEAR FLAG BYTES
   15  0025  30                    TSX
   16  0026  A6    00              LDA A   0,X             ;LOAD TOP OF STACK
   17  0028  81    00G             CMP A   EOLTG,I
   18  002A  26    06              BNE     1$
   19  002C  BD  019E'     5$:     JSR     DEFINT  ;NO ARGS COMPUTE DEFAULT INTERCEPTS INTO R12,R13
   20  002F  7E  0006'             JMP     XDRAW
   21  0032  81    00G     1$:     CMP A   ATSNTG,I        ;
   22  0034  27    F6              BEQ     5$
   23  0036  BD  0000G             JSR     SETARG          ;THERE MUST BE AT LEAST 2 VALID ARGS
   24  0039  BD  0000G             JSR     TYPARG
   25  003C  26    20              BNE     3$              ;THERE WEREN'T TWO VALID ARGUMENTS
   26  003E  A6    01              LDA A   1,X             ;2 VALID ARGUMENTS FOUND
   27  0040  81    00G             CMP A   EOLTG,I         ;IS EOL THERE INDICATING TWO ARGS ONLY?
   28  0042  26    0E              BNE     2$              ; 4 ARGUMENTS
   29  0044  30                6$: TSX
   30  0045  08                    INX
   31  0046  DF    00G             STX     R15,D           ;STORE POINTER TO Y TIC INTERVAL
   32  0048  BD  0000G             JSR     A9X
   33  004B  DF    00G             STX     R14,D           ;STORE POINTER TO X TIC INTERVAL
   34  004D  BD  019E'             JSR     DEFINT          ;GET DEFAULT INTERCEPT
   35  0050  20    26              BRA     TICTST
   36  0052  81    00G     2$:     CMP A   ATSNTG,'        ;IS THERE?(PHONY EOL)
   37  0054  27    EE              BEQ     6$
   38  0056  BD  0000G             JSR     CLRARG          ;4 ARGS BUT CHECK FOR VALIDITY
   39  0059  BD  0000G             JSR     TYPARG
   40  005C  27    07              BEQ     4$
   41  005E  86    00G     3$:     LDA A   ERDOMN,I
   42  0060  97    00G             STA A   ERRCD,D
   43  0062  7E  0192'             JMP     END
   44  0065  30                4$: TSX
   45  0066  08                    INX
   46  0067  DF    00G             STX     R13,D           ;STORE POINTER TO Y INTERCEPT
   47  0069  BD  0000G             JSR     A9X
   48  006C  DF    00G             STX     R12,D           ;STORE POINTER TO X INTERCEPT
   49  006E  BD  0000G             JSR     A9X
   50  0071  DF    00G             STX     R15,D           ;STORE POINTER TO Y TIC
   51  0073  BD  0000G             JSR     A9X
   52  0076  DF    00G             STX     R14,D           ;STORE POINTER TO X TIC
   53  0078  96    00G     TICTST: LDA A   R22,D
   54  007A  8A    11              ORA A   #11,I   ;SET X,Y TIC FLAGS
   55  007C  97    00G             STA A   R22,D
   56  007E  DE    00G             LDX     R14,D           ;GET POINTER TO X TIC
   57  0080  A6    00              LDA A   0,X             ; CLEAR SIGN BIT
```

```
0082   84     7F                AND A   127.,I
0084   A7     00                STA A   0,X
0086   EE     00                LDX     0,X                ;
0088   26     06                BNE     1$                 ;CHECK EXPONENT FOR ZERO
008A   86     EF                LDA A   H0EF,I
008C   94     00G               AND A   R22,D              ;CLEAR X TIC FLAG
008E   97     00G               STA A   R22,D
0090   DE     00G         1$:   LDX     R15,D              ;LOAD POINTER TO Y TIC
0092   A6     00                LDA A   0,X
0094   84     7F                AND A   127.,I
0096   A7     00                STA A   0,X
0098   EE     00                LDX     0,X                ;CHECK FOR ZERO
009A   26     06                BNE     2$
009C   86     FE                LDA A   H0FE,I ;CLEAR Y TIC FLAG
009E   94     00G               AND A   R22,D
00A0   97     00G               STA A   R22,D
00A2   BD     0000G       2$:   JSR     DOFP               ;GET Y INT +-1  FOR TICS ON Y AXIS
00A5   00G                      .BYTE   FHEX
00A6   0000G                    .WORD   YMAXW
00A8   A0G                      .BYTE   FHEX+FS
00A9   0000G                    .WORD   YMINW
00AB   C0G                      .BYTE   FHEX+FM
00AC   02EF'                    .WORD   FPP
00AE   16                       .BYTE   FDUP
00AF   00G                      .BYTE   FSAV
00B0   80G                      .BYTE   FHIN+FA
00B1   0000G                    .WORD   R13
00B3   00G                      .BYTE   FLIN
00B4   0000G                    .WORD   R16
00B6   00G                      .BYTE   FHIN
00B7   0000G                    .WORD   R13
00B9   A0G                      .BYTE   FRES+FS
00BA   00G                      .BYTE   FLIN
00BB   0000G                    .WORD   R17
00BD   00G                      .BYTE   FHEX               ;GET X INT +-1  FOR TICS ON X AXIS
00BE   0000G                    .WORD   XMAXW
00C0   A0G                      .BYTE   FHEX+FS
00C1   0000G                    .WORD   XMINW
00C3   C0G                      .BYTE   FHEX+FM
00C4   02CF'                    .WORD   FPP
00C6   16                       .BYTE   FDUP
00C7   00G                      .BYTE   FSAV
00C8   80G                      .BYTE   FHIN+FA
00C9   0000G                    .WORD   R12
00CB   00G                      .BYTE   FLIN
00CC   0000G                    .WORD   R18
00CE   00G                      .BYTE   FHIN
00CF   0000G                    .WORD   R12
00D1   A0G                      .BYTE   FRES+FS
]]]]]
00D2   00G                      .BYTE   FLIN
00D3   0000G                    .WORD   R19
00D5   00G                      .BYTE   FRET
```

```
                                        .SBTTL  XDRAW
                                ;       X DRAW
                                ;
0006  BD  01D9'         XDRAW:  JSR     INIMOV
0009  96  00G                   LDA A   R22,D           ;LOAD FLAG BYTE
0008  85  10                    BIT A   #10,I   ;X TIC SET?
000D  27  0F                    BEQ     DRAW1           ;X TIC IS NOT SET
000F  CE  0000G                 LDX     XMAXW,I
00E2  BD  0000G                 JSR     PSHFPN
00E5  CE  0020G                 LDX     SCRTCH+32.,I
00E8  BD  0000G                 JSR     PULFPN
00EB  BD  0205'                 JSR     XTICDR
00EE  CE  0000G         DRAW1:  LDX     XMAXW,I
00F1  BD  0000G                 JSR     PSHFPN
00F4  CE  0000G                 LDX     TEMPX,I
00F7  BD  0000G                 JSR     PULFPN
00FA  BD  02D7'                 JSR     GRFDRQ
00FD  BD  01D9'                 JSR     INIMOV
0100  96  00G                   LDA A   R22,D
0102  85  10                    BIT A   #10,I   ;X TIC SET?
0104  27  1C                    BEQ     DRAW2           ;X TIC IS NOT SET
0106  96  00G                   LDA A   R22,D
0108  8A  80                    ORA A   #80,I
010A  97  00G                   STA A   R22,D   ;SET "LEFT" FLAG
010C  CE  0000G                 LDX     XMINW,I
010F  BD  0000G                 JSR     PSHFPN
0112  CE  0020G                 LDX     SCRTCH+32.,I
0115  BD  0000G                 JSR     PULFPN
0118  DE  00G                   LDX     R14,D           ;LOAD X TIC INTERVAL
011A  A6  00                    LDA A   0,X
011C  8A  80                    ORA A   #80,I   ;SET TIC NEGATIVE
011E  A7  00                    STA A   0,X
0120  BD  0205'                 JSR     XTICDR
0123  CE  0000G         DRAW2:  LDX     XMINW,I
0126  BD  0000G                 JSR     PSHFPN
0129  CE  0000G                 LDX     TEMPX,I
012C  BD  0000G                 JSR     PULFPN
012F  BD  02D7'                 JSR     GRFDRQ
```

```
                                        .SBTTL  YDRAW
                                ;       YDRAWXXXXXXXXXXXXXXXX
0132  BD  01D9'    YDRAW:      JSR     'NIMOVE
0135  96  00G                 LDA A   R22,D           ;FLAGS
0137  84  11                  AND A   #011,I  ;CLEAR "DOWN" FLAG
0139  97  00G                 STA A   R22,D
013B  96  00G                 LDA A   R22,D
013D  47                      ASR A                   ;Y TIC FLAG SET?
013E  24  0F                  BCC     DRAW3           ;NO
0140  CE  0000G               LDX     YMAXW,I
0143  BD  0000G               JSR     PSHFPN
0146  CE  0020G               LDX     SCRTCH+32.,I
0149  BD  0000G               JSR     PULFPN  ;YMAX INTO TEMPY
014C  BD  026E'               JSR     YTICDR
014F  CE  0000G     DRAW3:    LDX     YMAXW,I
0152  BD  0000G               JSR     PSHFPN
0155  CE  0000G               LDX     TEMPY,I
0158  BD  0000G               JSR     PULFPN
015B  BD  02D7'               JSR     GRFDRQ
015E  BD  01D9'               JSR     INIMOV
0161  96  00G                 LDA A   R22,D
0163  47                      ASR A
0164  24  1C                  BCC     DRAW4
0166  96  00G                 LDA A   R22,D
0168  8A  80                  ORA A   #80,I   ;SET LOWER FLAG
016A  97  00G                 STA A   R22,D
016C  CE  0000G               LDX     YMINW,I
016F  BD  0000G               JSR     PSHFPN
0172  CE  0020G               LDX     SCRTCH+32.,I
0175  BD  0000G               JSR     PULFPN
0178  DE  00G                 LDX     R15,D           ;LOAD Y TIC INTERVAL
017A  A6  00                  LDA A   0,X
017C  8A  80                  ORA A   #80,I   ;SET TIC INTERVAL NEG
017E  A7  00                  STA A   0,X
0180  BD  026E'               JSR     YTICDR
0183  CE  0000G     DRAW4:    LDX     YMINW,I
0186  BD  0000G               JSR     PSHFPN
0189  CE  0000G               LDX     TEMPY,I
018C  BD  0000G               JSR     PULFPN
018F  BD  02D7'               JSR     GRFDRQ
0192  BD  01D9'     END:      JSR     INIMOV          ;MOVE TO INTERCEPT BEFORE EXITING
0195  BD  0000G               JSR     UNADR           ;RESET THE 170 SYSTEM
0198  BD  0000G               JSR     HALTR           ;CLEAN THE STACK
019B  7E  0000G               JMP     RTRN
```

```
    1                                           .SBTTL DEFINT   COMPUTE DEFAULT INTERCEPTS
    2  019E  FE  0000G     DEFINT: LDX   XMINW
    3  01A1  27  09                BEQ   XOINT
    4  01A3  2A  0C                BPL   XMNINT
    5  01A5  FE  0000G             LDX   XMAXW
    6
    7  01A8  27  0C                BEQ   XMXINT
    8  01AA  2B  05                BMI   XMNINT
    9  01AC  CE  0000G     XOINT:  LDX   FPZERO,I
   10  01AF  20  08                BRA   YDEF
   11  01B1  CE  0000G     XMNINT: LDX   XMINW,I
   12  01B4  20  03                BRA   YDEF
   13  01B6  CE  0000G     XMXINT: LDX   XMAXW,I
   14  01B9  DF  00G       YDEF:   STX   R12.D
   15  01BB  FE  0000G             LDX   YMINW   ;GET EXPONENT BYTES
   16  01BE  27  09                BEQ   YOINT
   17  01C0  2A  0C                BPL   YMNINT
   18  01C2  FE  0000G             LDX   YMAXW   ;GET EXPONENT BYTES
   19  01C5  27  0C                BEQ   YMXINT
   20  01C7  2B  05                BMI   YMNINT
   21  01C9  CE  0000G     YOINT:  LDX   FPZERO,I              ;SET YO INTERCEPT FLAG
   22  01CC  20  08                BRA   AFTER
   23  01CE  CE  0000G     YMNINT: LDX   YMINW,I         ;SET YMIN INTERCEPT FLAG
   24  01D1  20  03                BRA   AFTER
   25  01D3  CE  0000G     YMXINT: LDX   YMAXW,I
   26  01D6  DF  00G       AFTER:  STX   R13.D
   27  01D8  39                    RTS
   28                              ;
   29                              .SBTTL  INIMOV  ROUTINE TO MOVE TO INTERSECTION
   30  01D9  BD  0000G     INIMOV: JSR   DOFP
   31  01DC  00G                   .BYTE   FHIN
   32  01DD  0000G                 .WORD   R12             ;X INTERCEPT
   33  01DF  16                    .BYTE   FDUP
   34  01E0  00G                   .BYTE   FLEX
   35  01E1  0000G                 .WORD   TEMPX
   36  01E3  00G                   .BYTE   FHIN
   37  01E4  0000G                 .WORD   R13             ;Y INTERCEPT
   38  01E6  16                    .BYTE   FDUP
   39  01E7  00G                   .BYTE   FLEX
   40  01E8  0000G                 .WORD   TEMPY
   41  01EA  00G                   .BYTE   FRET
   42  01EB  96  00G               LDA A   A.SEC,D         ; GET NORMAL SEC. ADDRESS
   43  01ED  36                    PSH A                   ; SAVE IT
   44  01EE  86  15                LDA A   21.,I   ;MOVE
   45  01F0  BD  0000G             JSR     NEWSEC          ; ROUTINE TO SET UP NEW SEC. ADDR.
   46                              .GLOBL  NEWSEC
   47  01F3  BD  02D7'             JSR     GREDRQ
   48  01F6  32                    PUL A                   ; RESTORE OLD SEC. ADDR.
   49  01F7  BD  0000G             JSR     NEWSEC
   50                          ;COPY R12, R13 FROM THE STACK INTO TEMPX AND TEMPY
   51  01FA  BD  0000G             JSR     DOFP
   52  01FD  00G                   .BYTE   FLEX
   53  01FE  0000G                 .WORD   TEMPY
   54  0200  00G                   .BYTE   FLEX
   55  0201  0000G                 .WORD   TEMPX
   56  0203  00G                   .BYTE   FRET
   57  0204  39                    RTS
```

INIMOV ROUTINE TO MOVE TO INTERSECTION

```
                                
                        XTICOR: JSR     DOFP
                                .BYTE   FHEX
                                .WORD   TEMPX
                                .BYTE   FHIN+FA
                                .WORD   R14
                                .BYTE   FDUP
                                .BYTE   FLEX
                                .WORD   TEMPX
                                .BYTE   FHEX+FS
                                .WORD   SCRTCH+32.
                                .BYTE   FLEX
                                .WORD   SCRTCH+40.
                                .BYTE   FRET
                                LDX     SCRTCH+40.,I
                                LDA A   R22,D
                                ASL A
                                BCS     LEFT
                                LDA A   0,X
                                LDX     0,X
                                BEQ     DRAWIT
                                ASL A
                                BCC     RETI
                        DRAWIT: LDX     R13,D           ;Y INTERCEPT
                                JSR     PSHFPN
                                LDX     TEMPY,I
                                JSR     PULFPN
                                JSR     GRFDRQ
                                LDX     R16,D
                                JSR     PSHFPN
                                LDX     TEMPY,I
                                JSR     PULFPN
                                JSR     GRFDRQ
                                LDX     R17,D
                                JSR     PSHFPN
                                LDX     TEMPY,I
                                JSR     PULFPN
                                JSR     GRFDRQ
                                LDX     R13,D
                                JSR     PSHFPN
                                LDX     TEMPY,I
                                JSR     PULFPN
                                JSR     GRFDRQ
                                BRA     XTICOR
                        LEFT:   LDA A   0,X
                                LDX     0,X
                                BEQ     DRAWIT
                                ASL A
                                BCC     DRAWIT
                        RETI:   RTS
                        YTICOR: JSR     DOFP
                                .BYTE   FHEX
                                .WORD   TEMPY
                                .BYTE   FHIN+FA
                                .WORD   R15
                                .BYTE   FDUP
                                .BYTE   FLEX
```

```
                                                .WORD   TEMPY
                                                .BYTE   FHEX+FS
                                                .WORD   SCRTCH+32.
                                                .BYTE   FLEX
                                                .WORD   SCRTCH+40.
                                                .BYTE   FRET
                                                LDX     SCRTCH+40.,I
                                                LDA A   R22.D
                                                ASL A
                                                BCS     DOWN
                                                LDA A   0,X
                                                LDX     0,X
                                                BEQ     DRAZIT
                                                ASL A
                                                BCC     RET1
                                        DRAZIT: LDX     R12.D   ;X INTERCEPT
                                                JSR     PSHFPN
                                                LDX     TEMPX.I
                                                JSR     PULFPN
                                                JSR     GRFDRQ
                                                LDX     R18.D           ;X INT +
                                                JSR     PSHFPN
                                                LDX     TEMPX.I
                                                JSR     PULFPN
                                                JSR     GRFDRQ
                                                LDX     R19.D           ;X INT -
                                                JSR     PSHFPN
                                                LDX     TEMPX.I
                                                JSR     PULFPN
                                                JSR     GRFDRQ
                                                LDX     R12.D           ;X INT
                                                JSR     PSHFPN
                                                LDX     TEMPX.I
                                                JSR     PULFPN
                                                JSR     GRFDRQ
                                                BRA     YTICOR
                                        DOWN:   LDA A   0,X
                                                LDX     0,X
                                                BEQ     DRAZIT
                                                ASL A
                                                BCC     DRAZIT
                                                RTS
                                        GRFDRQ: JSR     DOFP
                                                .BYTE   FHEX
                                                .WORD   TEMPX
                                                .BYTE   FHEX
                                                .WORD   TEMPY
                                                .BYTE   FRET
                                                JSR     GRFDR.
                                                JSR     DOFP
                                                .BYTE   FLEX
                                                .WORD   YLAST
                                                .BYTE   FLEX
                                                .WORD   XLAST
                                                .BYTE   FRET
                                                RTS
                                        A3 FPP: .BYTE   03, HOFA, HOA3, HOD7
```

INIMOV ROUTINE TO MOVE TO INTERSECTION

```
                02F2    D7
172                     02E3    0A      3D      7D              .BYTE   H0A, H3D, H7D, H0A2
                02F6    A2
173                                                     ;ABOVE IS A FLOATING POINT .005 FOR TIC SIZE
174                             0001'                   .END
```

```
QAXIS  RT-11 MMAC  VM02-10   14-OCT-76 01:40:28 PAGE 5+
SYMBOL TABLE

ADRDEV= xxxxxx G      AFTER    01D6R      ATSNTG= xxxxxx G      A.SEC = xxxxxx G      A.STAT= xxxxxx G
A9X   = xxxxxx G      BERALC= xxxxxx G    CLRARG= xxxxxx G      DEFINT   019ER      DOFP  = xxxxxx G
DOWN     02CDR      DRAWIT   022AR      DRAW1    0GEER      DRAW2    0123R      DRAW3    014FR
DRAWN    0183R      DRAZIT   0293R      END      0192R      EOLTG = xxxxxx G      ERDOMN= xxxxxx G
ERRCD = xxxxxx G      FA    =  0080      FART  = xxxxxx G      FD    =  00E0      FDWP  =  0016
FHEX  = xxxxxx G      FHIN  = xxxxxx G    FLEX  = xxxxxx G      FLIN  = xxxxxx G      FM    =  00C0
FPCMP = xxxxxx G      FPP      02EFR      FPZERO= xxxxxx G      FRES  = xxxxxx G      FRET  = xxxxxx G
FS    =  0090      FSAV  = xxxxxx G    GREDRO   22D7R      GREDR = xxxxxx G      HALTR = xxxxxx G
INIMOV   01D9R      LEFT     0264R      NEWSEC= xxxxxx G      OPRADR= xxxxxx G      PLOTVL= xxxxxx G
PSHFPN= xxxxxx G      PSHRET= xxxxxx G    PULFPN= xxxxxx G      QAXIS    0000RG      RET1     026DR
RTRN  = xxxxxx G      R10   = xxxxxx G    R11   = xxxxxx G      R12   = xxxxxx G      R13   = xxxxxx G
R14   = xxxxxx G      R15   = xxxxxx G    R16   = xxxxxx G      R17   = xxxxxx G      R18   = xxxxxx G
R19   = xxxxxx G      R20   = xxxxxx G    R21   = xxxxxx G      R22   = xxxxxx G      R8    = xxxxxx G
R9    = xxxxxx G      SCRTCH= xxxxxx G    SETARG= xxxxxx G      TEMPX = xxxxxx G      TEMPY = xxxxxx G
TICTST   0078R      TYPARG= xxxxxx G    UNADR = xxxxxx G      XDRAW    0006R      XLAST = xxxxxx G
XMAXW = xxxxxx G      XMINS = xxxxxx G    XMINW = xxxxxx G      XMNINT   0181R      XMXINT   0186R
XNEW  = xxxxxx G      XTICDR   0205R      XOINT    01ACR      YDEF     01B9R      YDRAW    0132R
YLAST = xxxxxx G      YMAXW = xxxxxx G    YMINS = xxxxxx G      YMINW = xxxxxx G      YMNINT   01CER
YMXINT   01D3R      YNEW  = xxxxxx G    YTICDR   026ER      YOINT    01C9R      ZDRAW = xxxxxx G

. ABS.    0000     00
          02F7     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3072. WORDS
.SY:QAXIS/C(DK1:SEICL1,QAXIS
```

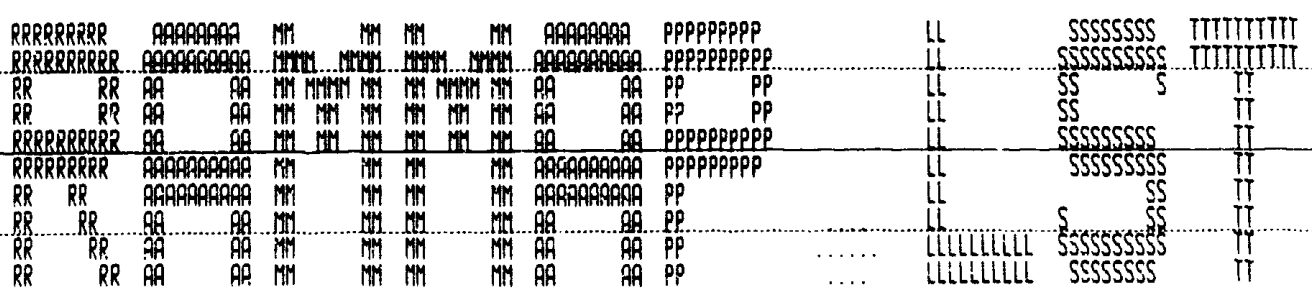
CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
R15     1-38#   2-31*   2-60*   2-65    4-31    5-112
R16     1-38#   2-3*    2-86    5-86
R17     1-38#   2-5*    2-91    5-91
R18     1-38#   2-7*    2-103   5-135
R19     1-38#   2-9*    2-108   5-140
R20     1-38#
R21     1-38#
R22     1-38#   1-39#   2-14*   2-53    2-55*   2-63    2-64*   2-72    2-73*   3-5     3-19    3-22    3-24*   4-4
        4-6*    4-7     4-21    4-24    4-26*   5-73    5-122
R8      1-38#
R9      1-38#
RET1    5-80    5-107#  5-129
RTRN    1-45#   4-44
SCRTCH  1-51#   2-2     2-4     2-6     2-8     3-10    3-27    4-12    4-29    5-68    5-70    5-72    5-117   5-119
        5-121
SETARG  1-43#   2-23
TEMPX   1-33#   3-15    3-36    5-35    5-55    5-61    5-66    5-132   5-137   5-142   5-147   5-159
TEMPY   1-34#   4-17    4-38    5-40    5-53    5-83    5-88    5-93    5-98    5-110   5-115   5-161
TICTST  2-35    2-53#
TYPARG  1-43#   2-24    2-39
UNADR   1-20#   4-42
XOINT   5-3     5-9#
XDRAW   2-20    3-4#
XLAST   1-42#   5-168
XMAXW   1-28#   2-93    3-8     3-13    5-5     5-13
XMINS   1-31#
XMINW   1-27#   2-95    3-25    3-34    5-2     5-11
XMNINT  5-4     5-8     5-11#
XMXINT  5-7     5-13#
XNEW    1-43#
XTICOR  3-12    3-33    5-59#   5-101
YOINT   5-16    5-21#
YDEF    5-10    5-12    5-14#
YDRAW   4-3#
YLAST   1-42#   5-166
YMAXW   1-30#   2-76    4-10    4-15    5-18    5-25
YMINS   1-32#
YMINW   1-29#   2-78    4-27    4-36    5-15    5-23
YMNINT  5-17    5-20    5-23#
YMXINT  5-19    5-25#
YNEW    1-43#
YTICOR  4-14    4-35    5-108#  5-150
ZDRAW   1-20#   2-10
```

CROSS REFERENCE TABLE (CREF V01-03)

SEL 1-3



```
                                .TITLE  RAMMAP  PAGE 0 AND ABOVE SPACE ALLOCATIONS
                                .IDENT  /RAM064/
                        ;
                                .GLOBL  PRTERR,PARSE,FPMATH,FDMAIN,ADBANK
                                .GLOBL  TRANSL,LEX,SYNTAB,SHUNT,GETLIN,INFO,PGMLIS
                                .GLOBL  DELETE,APPEND,COMPR,RENUMB,UNCOMP,UNLEX
                                .GLOBL  EVLTBL,EVLEN,PGMEVL,LDATA,FLCTRL,BSTMT,EVLSUB
                                .GLOBL  TYPIN,PWRUP,BNKMAP,XFMMAP,UTILS,MATOPR
                                .GLOBL  USING
                                .GLOBL  PRINTF,DIMSUB,STRING,CALL
                                .GLOBL  IOCTL,BINCTL,OUTCTL,INPCTL
                                .GLOBL  IOPROC,KBPROC
                                .GLOBL  DSPOUT,CLOSE,FILES
                                .GLOBL  INTSRV,IOKONS,GRAF,CROSS,AXIS
                                .GLOBL  CRTDRV,KEYDRV,IECDRV,MTDRV
                        ;
                                .GLOBL  TAPE,GRAPH
                                .GLOBL  GETKEY,PCHAR
                        ;
                                .ENABL  HEX
        000A                    .RADIX  10
                        ;
                                .MACRO  ASGN    ID,N
                                .GLOBL  ID
                        ID      =       N
                                =       N
                                .ENDM
                        ;
                                .MACRO  Z  ID,L
                                =       PGZ
                                .GLOBL  ID
                        ID      =       PGZ
                        PGZ     =       L+PGZ
                                .ENDM
                        ;
                                .MACRO  N  ID,L
                                =       PGN
                                .GLOBL  ID
                        ID      =       PGN
                        PGN     =       L+PGN
                                .ENDM
                        ;
        0000            .ASECT
                        .
        0000    PGZ     =       0
        0100    PGN     =       256
                        .
                        ;       STACK TAG DEFINATIONS
                        ;
  0000                  ASGN    NULLTG,0        ;NULL ENTRY
  0000                  ASGN    PLOSTG,1        ;POINTER TO LITERAL IN OBJECT STRING
  0001                  ASGN    ESTG,2          ;EVAL STATUS
  0002                  ASGN    GOSTG,3         ;GOSUB/RETURN DATA
  0003                  ASGN    FORTG,4         ;FOR/NEXT DATA
  0004                  ASGN    LISTTG,5        ;LIST COUNT FOR LIST COMMAND
  0005                  ASGN    PGMTG,6         ;POINTER TO PROGRAM
  0006                  ASGN    IMXTG,7         ;IMMEDIATE EXEC STATEMENT PTR
```

```
                                ASGN    PNTSTG,8        ;POINTER TO NAME TABLE STRING
                                ASGN    PSCTG,9         ;POINTER TO STRING COUNT
                                ASGN    PNTNTG,10       ;POINTER TO NAME TABLE NUMERIC
                                ASGN    PAETG,11        ;POINTER TO ARRAY ELEMENT
                                ASGN    VALTG,12        ;FLOATING POINT VALUE ON STACK
                                ASGN    LNNOTG,13       ;LINE NUMBER
                                ASGN    ITM1TG,14       ;2 BYTE ITEM - ONE
                                ASGN    ITM2TG,15       ;2 BYTE ITEM - TWO
                                ASGN    LBRKTG,16       ;LEFT BRACKET FOR DIMENSION
                                ASGN    SEMITG,17       ;SEMICOLON FOR PRINT
                                ASGN    ATSNTG,18       ;AT SIGN FOR I/O PACK
                                ASGN    BAKSTG,19       ;STACK CONTROL INFO FOR I/O PACK
                                ASGN    ALLTG,20        ;ALL IN DELETE ALL TAG
                                ASGN    RTRNTG,21       ;RETURN ADDRESS
                                ASGN    PRTTG,22        ;PRINT ENTRY - FROM PAGE FULL INTERRUPT
                                ASGN    CALLTG,23       ;CALL STATEMENT
                                ASGN    EOLTG,24        ;END OF LINE
                                ASGN    EOSTG,25        ;END OF STACK
                        ;
                        ;       STACK CONTROL CONSTANTS
                        ;
                                ASGN    FUDGH,1         ;NORMAL MIN SPACE
                                ASGN    FUDGL,244       ;THIS IS 500 IN TWO HALVES
                        ;
                                ASGN    SLOPH,0         ;MIN IN CALC MODE FOR EVAL AND LINE START
                                ASGN    SLOPL,100       ;THIS IS REAL MIN STACK SIZE DIFFERANCE
                        ;
                        ;       SYSTEM
                        ;
                                Z       R0,2            ;PSEUDO REGISTERS
                                Z       R1,2
                                Z       R2,2
                                Z       R3,2
                                Z       R4,2
                                Z       R5,2
                                Z       R6,2
                                Z       R7,2
                                Z       R8,2
                                Z       R9,2
                                Z       R10,2
                                Z       R11,2
                                Z       R12,2
                                Z       R13,2
                                Z       R14,2
                                Z       R15,2
                                Z       R16,2
                                Z       R17,2
                                Z       R18,2
                                Z       R19,2
                                Z       R20,2
                                Z       R21,2
                                Z       R22,2
                                Z       R23,2
                        ;
                        ;       ORDER IS IMPORTANT FOR I/O PACK
                        ;
                                N       KEYQUE,30       ; KEYBOARD KEY QUE
```

```
                                ASGN    QEND,KEYQUE+30  ; END OF QUE
                                N       TRASH,1         ;FOR I/O PACK
                                N       MTBFR,258       ;MAG TAPE BUFFER
                                N       EDTBFR,74       ;EDITOR BUFFER
                                N       IOBFR1,74       ;I/O BUFFER
                                N       SCRTCH,260      ;WORK AREA
                                ASGN    HCTRCS,SCRTCH+259 ;BACK END OF SCRATCH
                        ;
                        ;       ONTBL TO ENDTBL MUST BE IN ORDER (INIT IN BSTMT)
                        ;
                                N       ONTBL,16        ;ON UNIT LINES
                                N       EOFTBL,20       ;ON EOF UNIT LINES
                                N       FNTBL,52        ;USER FN LINES
                                N       KEYSTK,9        ;FN KEY STACK
                                ASGN    ENDTBL,PGN      ;END OF TABLES FOR INIT TO CLEAR
                        ;
                        ;       LEX
                        ;
                                Z       EGU,1           ;FLAG BITS
                                Z       CNT,1           ;GENERAL PURPOSE COUNTER
                                Z       FLAG,1          ;GENERAL PURPOSE FLAG BITS
                                Z       SHNTCT,2        ;BYTE COUNT FROM SHUNT
                                Z       TFLGS1,1        ;TRANSLATOR FLAGS
                                Z       LEXCNT,2        ;BYTE COUNT FROM LEX
                                Z       SCORE,1         ;USED BY UNCOMPILER
                                Z       MASK,1          ;BIT MASK FOR UNLEX
                                Z       COMP,1          ;COMPRESS FLAG FOR UNLEX
                                N       CRASH,10        ;HOLDING AREA FOR SYSTEM ERRORS
                                Z       STPTR,2         ;NAME TABLE PTR
                                Z       PGMPTR,2        ;PTR TO PROGRAM
                                Z       HEP,2           ;END OF RAM
                                Z       ZX,2            ;TO CLEAR INDEX REG
                                                        ;ZX MUST BE TWO BYTES AFTER PGMPTR
                                Z       SBP,2           ;STACK BOTTOM PTR
                                                        ;LSP & USRORG MUST BE IN ORDER (PWRUP)
                                Z       LSP,2           ;LOW STACK PTR
                                Z       USRORG,2        ;USER ADDRESS SPACE ORIGIN
                                Z       XEQSP,2         ;STACK POINTER FOR EXECUTION STACK
                                N       XEQSTK,16       ;STACK SPACE
                                Z       ERRCD,1         ;ERROR CODE
                                Z       ERRCDB,1        ;BACK UP AREA FOR ERRCD ON SIZE ERRORS
                                N       HLDEOF,1        ;HOLDING AREA FOR EOF ON UNIT TO PRTERR
                                N       ERCTRA,1        ;TOKEN COUNT AT ERROR IN PREFIX
                                N       ERCTRB,1        ;CHARACTER COUNT AT ERROR DISPLAY FORM
                        ;
                        ;       EVAL
                        ;
                                Z       CLPTR,2         ;CRNT LINE PTR
                                Z       NLPTR,2         ;NEXT LINE PTR
                                Z       NTPTR,2         ;NEXT TOKEN PTR
                                Z       CTKN,1          ;CRNT TOKEN
                                Z       LCLFLG,1        ;LOCAL FLAGS
                                Z       GLBFLG,1        ;GLOBAL FLAGS
                                Z       OPRADR,2        ;HOLDING AREA FOR OPERATOR ADDRESS
                                N       OPRRTN,2        ;TEMP RETURN HOLDING AREA
                                Z       DREXTA,3        ;EXIT JMP-A
                                Z       DREXTB,3        ;EXIT JMP-B
```

```
                                ;
0058                        N   CDSPTR,2        ;CRNT DATA STMT PTR
0439                        N   CDOPTR,2        ;CRNT DATA OBJ ENTRY PTR
0438                        N   BRKCNT,1        ;BRACKET COUNT FOR DIMENSION
                                ;
                                ;   JMPX, LDXX, LDAX, LDBX, STAX & JMPAX MUST BE IN ORDER (PWRUP)
                                ;
0430                        N   JMPX,7          ;FANCY JUMP
043E                        N   LDXX,6          ;FANCY LDX
0445                        N   LDAX,6          ;FANCY LDA
044B                        N   LDBX,6          ;FANCY LDA
0451                        N   STAX,6          ;FNACY STA
0457                        N   JMPAX,5         ;FANCIER JUMP
                                ;
                                ;   I/O & MONITOR
                                ;
0450                        Z   INIQQQ,2        ;  PLAYGROUND FOR INTERUPT SERVICE ROUTINE
005E                        Z   EDTEND,2        ;LAST TEXT LOCATION+1
0060                        Z   EDTPTR,2        ;CURRENT CURSOR FOR EDITOR
0062                        Z   EDTMAX,2        ;ADDR OF LAST CHAR IN BFR+1
0064                        Z   ANCRNT,2        ;CURRENT LINE NO. FOR AUTO NUMBER
0066                        Z   ANINCR,2        ;INCREMENT - IF = 0 AUTO NO. NOT ACTIVE
0068                        N   ANHCLO,1        ;NO AUTO NUMBER THIS ENTRY
0462                        Z   CRSTAT,1        ;CR STATUS  -  CR, EOFTYP OR EOITYP
006A                        Z   KBFLAG,1        ;KEYBOARD STATUS FLAGS
006B                        Z   LSTKEY,1        ;LAST VALID KEYCODE
006C                        Z   PNDFLG,1        ;INTERRUPT PENDING FLAG
006D                        Z   PNDEOF,1        ;FILE UNIT WITH EOF CONDITION
006E                        Z   MAGEND,1        ;  MAGTAPE HARDWARE FINISH FLAG
006F                        Z   MTSTT2,1        ;  SECOND MAGTAPE STATUS BYTE
0070                        Z   DSPSTT,1        ;  CRT STATUS BYTE
0071                        N   IECRLF,1        ;  CR VS  CR/LF FLAG
                                ;
                                ;   ORDER OF QIN AND QOUT FOR PWPUP
                                ;
0463                        Z   QIN,2           ;KEYQUE INPUT POINTER
0072                        Z   QOUT,2          ;KEYQUE OUTPUT POINTER
0074                        Z   KBIN,1          ;LAST KEYBOARD KEY DEPRESSED
0076                        Z   PPMODE,1        ;PERCENT MODE FLAGS
0077                        N   PPEOR,1         ;    OPTIONAL END OF LINE CHAR.
0464                        N   PPEOF,1         ;    END OF TEXT CHAR.
0465                        N   PPNUL,1         ;    NULL CHAR. (IGNORE THIS CHAR.)
0466                        N   IOSYSF,1        ;  I/O SYSTEM FIRST OUT - ROM PACK ONLY
0467                        N   MTMASK,1        ;  MAGTAPE MASK BYTE FOR ASCII TRANSFERS
0468                        Z   EOLCHR,1        ;  PRESENT END OF LINE CHAR.
0078                        Z   ETXCHR,1        ;  PRESENT END OF TEXT CHAR.
0079                        Z   NULCHR,1        ;  PRESENT NULL CHAR.  (IGNORE IF BIT 7 IS ON)
007A                        Z   STAT3?,,
0078                        Z   BANK,1          ;PRESENT BANK IN USE
007C                        N   XFNBNK,4        ;ADDR OF EXTENDED FUNCTION ROM PACK
0469                        Z   LUNNO,1         ;LOGICAL UNIT NUMBER FOR FILE
0070                        Z   RECNO,2         ;RECORD NUMBER FOR FILE
007E                        Z   XAXIS,1
0080                        N   DSPVCT,3        ;  TERMINAL MODE VECTOR
                                                ;  IN THE CHARACTER GEN LOOP
046D                        N   IDLVCT,3        ;  TRANSFER VECTOR IN THE IDLE LOOP
                                                ;  2 BYTES LOCATION
```

```
244                                                     ; 1 BYTE BANK
245     0470                            N       CHRVCT,3        ; INTERCEPT CHAR GEN CODE
246     0473                            Z       MTSREG,1        ; MAGTAPE SET STATUS REG. (256/128, ETC.)
247     0081                            Z       MTSTAT,1        ;MAG TAPE STATUS
248     0082                            Z       MTMAX,2         ; MAG TAPE MAX. BUFFER POINTER
249     0083                            Z       MTPTR,2         ;MAG TAPE BUFFER POINTER
250     0085                            Z       NOOUT,1         ;FLAGS FOR OUTPUT CONTROL
251     0087                            Z       TAGPTR,2        ;I/O LIST STACK POINTER
252     0088                            Z       IOFLGS,1        ;I/O PROCESSOR FLAGS
253     008A                            Z       CHRCNT,2        ;ACTUAL CHARACTER COUNT
254     008B                            N       POINTB,2
255     0476                            Z       CHRW,1          ;LAST CHARACTER TRANSFERRED UNDER I/O
256     008D                            Z       IOFUNC,1        ;PRESENT I/O FUNCTION
257     008E                            Z       C.STAT,1        ;CHANNEL 2 STATUS
258     008F                            Z       A.PRIM,1        ;CHAN. A PRIMARY ADDRES
259     0090                            Z       A.STRT,2        ;START OF CHAN. A BUFFER
260     0091                            Z       A.MAX,2         ;MAXIMUM AVAILABLE CHAR. POSITION
261     0093                            Z       A.END,2         ;LAST VALID CHAR. IN BUFFER+1
262     0095                            Z       A.SEC,1         ;CHAN. A SECONDARY ADDRESS
263     0097                            Z       A.PTR,2         ;POINTER TO CHAN. A BUFFER
264     0098                            ASGN    INPNT,A.PTR     ;PATCH FOR NOW TO RESOLVE NAMES
265     0098                            Z       LENGTH,2        ;LENGTH ATTRRIBUTE IN THE I/O PROCESSOR
266     009A                            Z       POINT,2         ;POINTER TO PRESENT VALUE
267     009C                            Z       COLCNT,2        ;MATRIX COL. COUNT
268     009E                            Z       TABPTR,2        ;POINTER INTO NAME TABLE
269     00A0                            Z       TCOL,2          ;TEMP. OF COLCNT
270     00A2                            N       EDTFLG,1        ;FOR COM PACK
271     0478                            Z       YAXIS,1         ;INTERNAL GIN CONTROL
272     00A4                            N       SCNSED,1        ;
273
274                             ;       XMINW TO TEMPY MUST BE IN ORDER - GRAF
275
276     0479                            N       XMINW,8         ;WINDOW (USERS UNITS)
277     047A                            N       XMAXW,8
278     0482                            N       YMINW,8
279     048A                            N       YMAXW,8
280     0492                            N       DLTXS,8         ;VIEWPORT IN GDUS
11111
281     049A                            N       XMINS,8
282     04A2                            N       DLTYS,8
283     04AA                            N       YMINS,8
284     0492                            N       XSF,8           ;AXIS SCALE FACTOR (USERS UNITS/GDU)
285     048A                            N       XLAST,8         ;POSITION OF LAST POINT PLOTTED
286     04C2                            N       YSF,8
287     04CA                            N       YLAST,8
288     04D2                            N       SINTHA,8        ;RELATIVE ROTATE CONSTANT
289     04DA                            N       COSTHA,8
290     04E2                            N       XNEW,8          ;NEW POSITION FOR PLOTTING
291     04EA                            N       TEMPX,8
292     04F2                            N       YNEW,8
293     04FA                            N       TEMPY,8
294     0502                            N       MSKCTR,1        ;DISABLE INTERRUPT COUNTER
295     050A                            N       ANYINT,1        ;NOT ZERO IF ANY INTERRUPTS PENDING
296     0508                            N       PIATBL,6*21     ;TABLE OF PIA ADDRESSES
297     050C                            N       PIAEND,2        ; END OF TABLE MARKER !!!
298                                                             2 BYTES - PIA ADDRESS
299                             ;                               1 BYTE - BANK ADDRESS
300                             ;                               2 BYTES - INTERRUPT SERVICE ROUTINE ADDRESS
```