```
                                        .SBTTL RPLOT ROTATION OF RELATIVE VECTORS
                                        ; DY, DX ON THE STACK
                                        ; EXIT WITH Y, X VALUE IN USER'S UNITS ON THE STACK

042C                            RPLOT:

                                        ; XVALUE = XLAST +(DX((COS THETA) - DY(SIN THEATA))

042C    BD    0000G                     JSR     DOFP
042F    00G                             .BYTE   FHIN
0430    0000G                           .WORD   TABPTR
0432    C0G                             .BYTE   FHEX+FM
0433    0000G                           .WORD   COSTHA
0435    00G                             .BYTE   FHEX
0436    0000G                           .WORD   SINTHA
0438    C0G                             .BYTE   FHIN+FM
0439    0000G                           .WORD   COLCNT
043B    A0G                             .BYTE   FART+FS
043C    80G                             .BYTE   FHEX+FA
043D    0000G                           .WORD   XLAST
043F    00G                             .BYTE   FLEX
0440    0000G                           .WORD   XNEW

                                        ; Y VALUE=+(DX(SIN THETA)+DY((COS THETA))

0442    00G                             .BYTE   FHIN
0443    0000G                           .WORD   TABPTR
0445    C0G                             .BYTE   FHEX+FM
0446    0000G                           .WORD   SINTHA
0448    00G                             .BYTE   FHIN
0449    0000G                           .WORD   COLCNT
044B    C0G                             .BYTE   FHEX+FM
044C    0000G                           .WORD   COSTHA
044E    80G                             .BYTE   FART+FA
044F    80G                             .BYTE   FHEX+FA
0450    0000G                           .WORD   YLAST
0452    00G                             .BYTE   FLEX
0453    0000G                           .WORD   YNEW
0455    00G                             .BYTE   FRET
0456    39                      1$:     RTS
```

```
                        .SBTTL ------------------------- -------------------------
                        .SBTTL CLIPPING ROUTINES FOR DRAW (AND RDRAW)
                ;*****************************************************************************
                ;
                ;       ROUTINES TO CLIP
                ;
                ;*****************************************************************************
                ;
                ; THESE ARE THE CLIP CODES:  (0000 IS THE VIEWPORT AREA)
                ;                 x       x
                ;        1001   x 1000  x 1010
                ;                 x       x
                ;       xxxxxxxxxxxxxxxxxxxxxxxx
                ;                 x       x
                ;        0001   x 0000  x 0010
                ;                 x       x
                ;       xxxxxxxxxxxxxxxxxxxxxxxx
                ;                 x       x
                ;        0101   x 0100  x 0110
                ;                 x       x
                ;
                ; DRAW WILL SET THE CLIP CODES OF NEW AND OLD POINTS
                ;       A DECISION WILL BE MADE ON TRIVIAL DRAW AND
                ;       NO DRAW SITUATIONS BY LOOKING AT CLIP CODE.  NEXT
                ;       CLIPIT IS CALLED TO TRIM VECTOR TO THE LIMITS
                ;       OF THE WINDOW.
                ;
                ; XNEW & YNEW HAVE AT FIRST THE NEW POINT TO BE PLOTTED BUT MAY BE
                ;       MODIFIED TO BE AT THE POINT WHERE
                ;       THE VECTOR CROSSES THE WINDOW BOUNDS.
                ;
                ; XLAST & YLAST HAVE AT FIRST THE LAST POINT PLOTTED,
                ;       BUT MAY BE MODIFIED TO BE POINT WHERE THE
                ;       VECTOR CROSSES THE WINDOW.
                ;
                ; TEMPX & TEMPY HAVE THE NEW POINT.  USED TO SAVE
                ;       THE POINT AND STORE IN X & YLAST UPON
                ;       COMPLETION OF DRAW ROUTINE.
                ; STACK AT ENTRY Y VALUE, X VALUE
                ;
                ; WILL CHANGE XNEW, YNEW, XLAST, YLAST, TEMPX, TEMPY, R0-R7
                ;
```

```
                                        .SBTTL DRAW SUBROUTINE FOR DRAW
0457  BD  0000G     GRFDRW: JSR   DOEP
045A  00G                   .BYTE  FHIN          ; SET UP TEMPY
0458  0000G                 .WORD  COLCNT
045D  00G                   .BYTE  FLEX
045E  0000G                 .WORD  TEMPY
0460  00G                   .BYTE  FHIN    ; SET UP TEMPX
0461  0000G                 .WORD  TABPTR
0463  00G                   .BYTE  FLEX
0464  0000G                 .WORD  TEMPX
0466  00G                   .BYTE  FRET
0467  CE  0000G     GRFDR.: LDX   XLAST,I ; GET CLIP CODE FOR LAST AND STORE IN SECOND BYTE
                                          ; OF R10
046A  DF  00G               STX   R1,D           ; R1 IS POINTER TO LAST POINT
046C  BD  0513'             JSR   CLIP           ; GET CLIP CODE FOR LAST
046F  D7  01G               STA B R10+1,D        ; CODE STORED SECOND BYTE OF R10
0471  CE  0000G             LDX   TEMPX,I ; GET CLIP CODE FOR NEW AND STORE IN FIRST BYTE
                                          ; OF R10
0474  DF  00G               STX   R1,D
0476  BD  0513'             JSR   CLIP
0479  D7  00G               STA B R10,D
047B  26  04                BNE   2$       ; TEST TRIVIAL DRAW, BOTH CLIPCODES=0
047D  96  01G               LDA A R10+1,D
047F  27  65                BEQ   APLOT          ; BRANCH BOTH=0
                                          ; TEST TRIVIAL NO DRAW, THAT IS
0481  D4  01G     2$:       AND B R10+1,D        ; NEW CLIP AND OLD CLIP <>0
0483  26  74                BNE   NOPLOT
                                          ; CLIP FROM NEW POINT TO VIEWPORT
0485  D6  00G               LDA B R10,D          ; CLIP CODE FOR NEW POINT
0487  27  19                BEQ   3$             ; BRANCH NO CLIP NECESSARY
0489  CE  0000G             LDX   TEMPX,I
048C  DF  00G               STX   R1,D           ; R1 POINTS TO XNEW AND YNEW
048E  CE  0000G             LDX   XLAST,I
0491  DF  00G               STX   R2,D           ; R2 POINTS TO XLAST AND YLAST
0493  86  02                LDA A 2,I            ; GET MAX. ITERATION COUNT
0495  36            4$:    PSH A
0496  BD  0572'             JSR   CLIPIT
0499  32                   PUL A
049A  5D                    TST B                ; CLIP COMPLETE
049B  27  05                BEQ   3$
049D  4A                    DEC A
049E  26  F5                BNE   4$
04A0  20  57                BRA   NOPLOT
04A2                3$:                      ; CLIP FROM LAST POINT TO VIEWPORT
04A2  97  00G               STA A R0,D           ; SAVE 2-# OF CLIPS
04A4  D6  01G               LDA B R10+1,D        ; CLIP CODE FOR LAST POINT
04A6  27  3E                BEQ   APLOT          ; BRANCH NO CLIP NECESSARY
04A8  CE  0000G             LDX   TEMPX,I
04AB  DF  00G               STX   R2,D           ; R2 POINTS TO XNEW AND YNEW
04AD  CE  0000G             LDX   XLAST,I
0480  DF  00G               STX   R1,D           ; R1 POINTS TO XLAST AND YLAST
04B2  86  02                LDA A 2,I            ; GET MAX. ITERATION COUNT
04B4  36            5$:    PSH A
0485  BD  0572'             JSR   CLIPIT         ; TRY TO CLIP THIS ONE
04B8  32                    PUL A
04B9  5D                    TST B                ; CLIP COMPLETE?
04BA  27  05                BEQ   1$
```

```
    58          04BC    4A                          DEC A
    59          04BD    26    F5                    BNE    5$
    60          04BF    20    38                    BRA    NOPLOT
    61          04C1    81    02          1$:       CMP A  2,I            ; TEST # OF CLIPS
    62          04C3    27    04                    BEQ    PLTOK
    63          04C5    91    00G                   CMP A  R0,D           ; HERE ACC. A MUST =1
    64          04C7    27    30                    BEQ    NOPLOT
    65          04C9    96    00G         PLTOK:    LDA A  A.SEC,D        ;TEMP STORAGE OF IEC SECONDARY ADDR
    66          04CB    36                          PSH A
    67          04CC    86    15                    LDA A  21.,I          ; SECONDARY ADDRESS SET FOR
    68                                                                    ; DEFAULT MOVE
    69          04CE    8D    3A                    BSR    NEWSEC         ; RESET IEC I/O AND NEW SEC. ADDR.
    70          04D0    BD    0000G                 JSR    DOFP           ; MOVE TO THE NEW XLAST,YLAST
    71          04D3                                SET    XLAST,XNEW
        04D3      00G                       .BYTE  FHEX
        04D4    0000G                       .WORD  XLAST
        04D6      00G                       .BYTE  FLEX
        04D7    0000G                       .WORD  XNEW
    72          04D9                                SET    YLAST,YNEW
        04D9      00G                       .BYTE  FHEX
        04DA    0000G                       .WORD  YLAST
        04DC      00G                       .BYTE  FLEX
        04DD    0000G                       .WORD  YNEW
    73          04DF    00G                         .BYTE  FRET
    74          04E0    BD    03A8'                 JSR    PLOTVL
    75          04E3    32                          PUL A                 ;RESTORE IEC
    76          04E4    8D    24                    BSR    NEWSEC
    77          04E6    BD    0000G       APLOT:    JSR    DOFP
    78          04E9                                SET    TEMPX,XNEW     ; MOVE TO TEMPX,TEMPY
        04E9      00G                       .BYTE  FHEX
        04EA    0000G                       .WORD  TEMPX
        04EC      00G                       .BYTE  FLEX
        04ED    0000G                       .WORD  XNEW
    79          04EF                                SET    TEMPY,YNEW
        04EF      00G                       .BYTE  FHEX
        04F0    0000G                       .WORD  TEMPY
        04F2      00G                       .BYTE  FLEX
        04F3    0000G                       .WORD  YNEW
    80          04F5    00G                         .BYTE  FRET
    81          04F6    BD    03A8'                 JSR    PLOTVL  ; PLOT TO XNEW, YNEW
    82          04F9    BD    0000G       NOPLOT:   JSR    DOFP
    83          04FC    00G                         .BYTE  FHIN           ; SET XLAST,YLAST
    84          04FD    0000G                       .WORD  T%BPTR
    85          04FF      00G                       .BYTE  FLEX
    86          0500    0000G                       .WORD  XLAST
    87          0502      00G                       .BYTE  FHIN
    88          0503    0000G                       .WORD  COLCNT
    89          0505      00G                       .BYTE  FLEX
]]]]]
    90          0506    0000G                       .WORD  YLAST
    91          0508      00G                       .BYTE  FRET
    92          0509    39                          RTS
```

```
                                        .SBTTL  NEWSEC--- ESTABLISH NEW SEC. ADDR. AND FLUSH OUTPUT BUFFER
                                ;THIS ROUTINE IS A RS1 PATCH
                                ;IT WILL FLUSH THE CURRENT OUTPUT BUFFER AND ESTABLISH A NEW SEC. ADDR.
                                ;FOR USE IN ENSUING TRANSFERS.  THIS IS SUCH THAT CLIPPING AND AXIS
                                ;WILL WORK CORRECTLY.
                                ;
                                ; CALL WITH NEW SEC. ADDR. IN ACC A
                                ;
                                        .GLOBL  NEWSEC
                                        .GLOBL  OUTBFR
0500    BD      0000G   NEWSEC: JSR     OUTBFR          ; OUTBFR SAVES ACC A
0503    97      00G             STA A   A.SEC,D         ; SNEAKY YES
0505    7F      0000G           CLR     A.STAT          ; THAT'S THE MAGIC LINE
0508    39                      RTS
```

```
                                        .SBTTL CLIP SUBROUTINE TO GENERATE THE CLIP CODE
                                               ; R1 POINTS TO X, Y PAIR ON ENTRY
                                               ; B ACCUMULATOR HAS CLIP CODE ON EXIT
                                        .GLOBL FPCMP
                                        .GLOBL A9X
                                        ;
0513  9F  00G     CLIP:  STS    R11,D           ; MARK STACK POINTER
0515  DE  00G            LDX    R1,D            ; LOOK AT X BOUNDRIES
0517  BD  0000G          JSR    PSHFPN
051A  CE  0000G          LDX    XMINW,I
051D  BD  0000G          JSR    PSHFPN
0520  BD  0000G          JSR    FPCMP           ; TEST THEM
0523  5F                CLR B                  ; INITIALIZE B
0524  4D                TST A
0525  2C  04             BGE    1$              ; BRANCH X=>XMINW
0527  C8  01             EOR B  1,I
0529  20  15             BRA    YTST
0528  30           1$:  TSX                     ; CLENA OLD STUFF OFF STACK
052C  BD  0000G          JSR    A9X
052F  35                 TXS
0530  CE  0000G          LDX    XMAXW,I
0533  BD  0000G          JSR    PSHFPN
0536  32                PUL A
0537  37                PSH B                  ; UPDATE B
0538  BD  0000G          JSR    FPCMP
0538  33                PUL B
053C  2F  02             BLE    YTST            ; BRANCH X<=XMAXW
053E  C8  02             EOR B  2,I
0540  DE  00G     YTST:  LDX    R1,D            ; LOOK AT Y BOUNDARIES
0542  BD  0000G          JSR    A16X
0545  BD  0000G          JSR    PSHFPN
0548  CE  0000G          LDX    YMINW,I
054B  BD  0000G          JSR    PSHFPN
054E  32                 PUL A
054F  37                 PSH B                  ; SAVE B SOMEWHERE
0550  BD  0000G          JSR    FPCMP
0553  33                PUL B
0554  2C  04             BGE    2$              ; BRANCH IF Y => YMINW
0556  C8  04             EOR B  4,I
0558  20  15             BRA    CLIPDN
055A  30           2$:  TSX                     ; CLENA UP OLD STUFF
055B  BD  0000G          JSR    A8X
055E  35                 TXS
055F  CE  0000G          LDX    YMAXW,I
0562  BD  0000G          JSR    PSHFPN
0565  32                 PUL A
0566  37                 PSH B
0567  BD  0000G          JSR    FPCMP
056A  33                 PUL B                  ; RESTORE CLIP CODES
056B  2F  02             BLE    CLIPDN          ; BRANCH IF Y <= YMAXW
056D  C8  08             EOR B  10,I
056F  9E  00G     CLIPDN: LDS    R11,D           ; RESET STACK POINTER
0571  39                 RTS
```

```
                                        .SBTTL CLIPIT ROUTINE TO CALCULATE BOUNDRY INTERSECTION
                                        ; CLIPIT ROUTINE CALCULATES THE
                                                ; INTERSECTION OF THE VECTOR AND
                                                ; THE WINDOW BOUNDRIES
                                        ; CLIP IS MADE FROM THE X - Y END POINT,
                                                ; POINTED TO BY R1, TO THE VIEWPORT.
                                                ; THIS X - Y POINT WILL BE CHANGED, IF NECESSARY,
                                                ; TO A POINT ON THE WINDOW LIMITS
                                        ; R2 POINTS TO THE OTHER X - Y ENDPOINT.
                                                ; THIS X - Y POINT WILL NOT BE CHANGED
                                                ; BY THIS ROUTINE.
                                        ; ENTRY WITH R1 & R2 POINTING TO X - Y VALUES
                                                ; AND B ACCUM HAVING CLIP CODE.
                                        ; R3, R4, R5, R6, AND R7 WILL BE CLOBBERED
                                        ; XNEW & YNEW REFER TO X & Y VALUES POINTED TO BY R1
                                                ; X2 & Y2 REFER TO X & Y VALUES POINTED TO BY R2
0572  C5  01        CLIPIT: BIT B   1,I             ; TEST LESS THAN XMINW
0574  27  05                BEQ     1$
                                                ; CLIP TO XMINW
                                                ; YNEW=YNEW+(Y2+YNEW)*(XMINW-XNEW)/(X2-XNEW)
                                                ; XNEW=XMINW
0576  CE  0000G             LDX     XMINW,I
0579  20  07                BRA     XDEFN           ; R7 WILL POINT TO XMINW
0578  C5  02        1$:     BIT B   2,I
057D  27  19                BEQ     YTEST
                                                ; CLIP TO XMAXW
                                                ; YNEW=YNEW+(Y2-YNEW)*(XMAXW-XNEW)/(X2-XNEW)
                                                ; XNEW=XMAXW
057F  CE  0000G             LDX     XMAXW,I         ; R7 WILL POINT TO XMAXW
0582  DF  00G       XDEFN:  STX     R7,D
0584  DE  00G               LDX     R1,D
0586  DF  00G               STX     R4,D            ; R4 POINTS TO XNEW
0588  BD  0000G             JSR     A16X
058B  DF  00G               STX     R3,D            ; R3 POINTS TO YNEW
058D  DE  00G               LDX     R2,D
058F  DF  00G               STX     R6,D            ; R6 POINTS TO X2
0591  BD  0000G             JSR     A16X
0594  DF  00G               STX     R5,D            ; R5 POINTS TO Y2
0596  20  25                BRA     EVAL
0598  C5  04        YTEST:  BIT B   4,I
059A  27  05                BEQ     1$
                                                ; CLIP TO YMINW
                                                ; XNEW=XNEW+(X2-XNEW)*(YMAXW-YNEW)/(Y2-YNEW)
                                                ; YNEW=YMAXW
059C  CE  0000G             LDX     YMINW,I         ; R7 WILL POINT TO YMINW
059F  20  08                BRA     YDEFN
05A1  C5  08        1$:     BIT B   10,I
05A3  26  01                BNE     YDEF.
05A5  39                    RTS
                                                ; CLIP TO YMAXW
                                                ; XNEW=XNEW+(X2-XNEW)*(YMAXW-YNEW)/(Y2-YNEW)
                                                ; YNEW=YMAXW
05A6  CE  0000G     YDEF.:  LDX     YMAXW,I         ; R7 WILL POINT TO YMAXW
05A9  DF  00G       YDEFN:  STX     R7,D
05AB  DE  00G               LDX     R1,D
05AD  DF  00G               STX     R3,D            ; R3 POINTS TO XNEW
05AF  BD  0000G             JSR     A16X
```

```
                                                STX     R4,D            ; R4 POINTS TO YNEW
                                                LDX     R2,D
                                                STX     R5,D            ; R5 POINTS TO X2
                                                JSR     A16X
                                                STX     R6,D            ; R6 POINTS TO Y2
                                                ; EVALUATE s3=s3+(s5-s3)*(s7-s4)/(s6-s4)
                                                ; AND s4=s7
                                        EVAL:   JSR     DOFP
                                                .BYTE   FHIN
                                                .WORD   R4
                                                .BYTE   FHIN+FS
                                                .WORD   R7
                                                .BYTE   FHIN
                                                .WORD   R4
                                                .BYTE   FHIN+FS
                                                .WORD   R6
                                                .BYTE   FART+FD
                                                .BYTE   FHIN
                                                .WORD   R5
                                                .BYTE   FHIN+FS
                                                .WORD   R3
                                                .BYTE   FART+FM
                                                .BYTE   FHIN+FA
                                                .WORD   R3
                                                .BYTE   FLIN
                                                .WORD   R3
                                                .BYTE   FHIN
                                                .WORD   R7
                                                .BYTE   FLIN
                                                .WORD   R4
                                                .BYTE   FRET
                                                JSR     CLIP
                                                RTS

                                                .END
```

SYMBOL TABLE

```
ABRFLG=  0040        ADDAIX   0205R       ADDARG   0203R       ADRDEV= XXXXXX G     AFAIL =  0030
AMAT  =  0010        APLOT    04E6R       ARRAY =  0020        ASTR  =  0020        ATLOD =  0008
ATSNTG= XXXXXX G     ATVLD =  0004        AXIERR   018FR       AXISCA   0196R       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     A16X  = XXXXXX G     A8X   = XXXXXX G     A9X   = XXXXXX G     A9X   = XXXXXX G
BAKSTG= XXXXXX G     BANK  = XXXXXX G     BFRALC= 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     CLIP     0513R       CLIPON   056FR
CLIPIT   0572R       CLPTR = XXXXXX G     CLRARG= XXXXXX G     CMAT  =  0001        COLCNT= XXXXXX G
COSCOD= XXXXXX G     COSTHA= XXXXXX G     CRDC3 =  0002        CREOF =  0008        CREOI =  0004
CREOT =  0020        CRETX =  0010        CRLF  = XXXXXX G     CRNORM=  0001        CRSTAT= XXXXXX G
CRVLD =  0080        CSTR  =  0002        CTKN  = XXXXXX G     CURSOR= XXXXXX G     DATDEV=  0022
DIMFLG=  0004        DIRCT =  0080        DISCNT= XXXXXX G     DISSRQ=  0008        DL    = XXXXXX G
DLTXS = XXXXXX G     DLTYS = XXXXXX G     DOFP  = XXXXXX G     DP    = XXXXXX G     DREXTA= XXXXXX G
DREXTB= XXXXXX G     DSPDEV=  0020        DT    = XXXXXX G     EDTBFR= XXXXXX G     ENDKEY=  0040
EOFTYP=  0038        EOLTG = XXXXXX G     EOSTG = XXXXXX G     ERATSN= XXXXXX G     ERDOMN= XXXXXX G
EREOM = XXXXXX G     EREBFR= XXXXXX G     ERFILE= XXXXXX G     ERIOE = XXXXXX G     ERNLOD= XXXXXX G
ERNSEP= XXXXXX G     ERRCD = XXXXXX G     EXTERM= XXXXXX G     ERUNDF= XXXXXX G     ESTG  = XXXXXX G
EVAL     058DR       EXTFLG=  0080        FA    =  0080        FART  = XXXXXX G     FD    =  00E0
FDUP  =  0016        FHEX  = XXXXXX G     FHIN  = XXXXXX G     FILDEV=  0000        FIX1A =  0003
FIX1B =  0004        FLEX  = XXXXXX G     FLIN  = XXXXXX G     FM    =  00C0        FMTVLD=  0008
FNFLG =  0010        FORTG = XXXXXX G     FPA   = XXXXXX G     FPADD = XXXXXX G     FPB   = XXXXXX G
FPC   = XXXXXX G     FPCMP = XXXXXX G     FPDIV = XXXXXX G     FPMUL = XXXXXX G     FPONE = XXXXXX G
FPSUB = XXXXXX G     FPZERO= XXXXXX G     FRES  = XXXXXX G     FRET  = XXXXXX G     FS    =  00A0
FSAV  = XXXXXX G     FSWP  = XXXXXX G     GETIT    011CR       GINSET   02EBRG      GINSTK   0309R
GLBFLG= XXXXXX G     GOSTG = XXXXXX G     GRAF  =  0000RG      GRAFER   0385R       GRAFRT   0389R
GRAPH    0236RG      GRFAGN   026AR       GRFDRW   0457R       GRFDR.   0467RG      GRFED    0284R
GRFEXT   029ER       GRFGIN   028JR       GRFINI   0000RG      GRFMOV   038CR       GRFNRM   02ABR
GRFREL   02AER       GRFSCN   035DR       GSTKEO   0346R       GSTKEX   034BR       IMXTG = XXXXXX G
INPUTK=  0001        INPVAL= XXXXXX G     IOBFR1= XXXXXX G     IOCLNR= XXXXXX G     IOFLGS= XXXXXX G
IOFUNC= 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
K.100    0047R       K.130    004FR       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        MATSIZ= XXXXXX G     MTBFR = XXXXXX G     MTPDEV=  0021
MTPD2 =  0023        NEWSEC   050ARG      NLPTR = XXXXXX G     NOKEY =  0080        NOOUT = XXXXXX G
NOPLOT   04F9R       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
NULLTG= XXXXXX G     OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000
ONSFLG=  0002        OPRADR= XXXXXX G     OUTBFR= XXXXXX G     PAETG = XXXXXX G     PARM  =  0008
PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMFP =  0003        PGMLEN=  0000
PGMLNM=  0007        PGMPTR= XXXXXX G     PGMTG = XXXXXX G     PLOSTG= XXXXXX G     PLOTVL   03A8RG
PLTIT    0302R       PLTOK    04C9R       PNDEOF= XXXXXX G     PNDFLG= XXXXXX G     PNTNTG= XXXXXX G
PNTSTG= XXXXXX G     POINT = XXXXXX G     PPMODE= XXXXXX G     PRIDEF=  0001        PRIVAL= XXXXXX G
PRTTG = XXXXXX G     PSCTG = XXXXXX G     PSHFPN= XXXXXX G     PSHRET= XXXXXX G     PULFPN= XXXXXX G
QDRAW    0213RG      QGIN     022FRG      QMOVE    0221RG      QRDRAW   021ARG      QRMOVE   0228RG
QROTAT   0057RG      QSCALE   0174RG      QVIEW    012ERG      QWINDO   00CARG      QWINEX   0113R
RECLFG=  0004        RELDRW   03E2R       RELMOV   0415R       RFAIL =  00C0        RMAT  =  0040
RPLOT    042CR       RPTCTL=  0080        RSTR  =  0080        RTRN  = XXXXXX G     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     SCALEF   0088R       SCALER=  0040        SCL   =  0001
SECDEF=  0002        SEMITG= XXXXXX G     SETARG= XXXXXX G     SINCOD= XXXXXX G     SINTHA= XXXXXX G
```

```
GRAF ----THIS IS REALLY GRAPHIC       RT-11 MMAC  VM02-10   14-OCT-76 01:33:37 PAGE 23+
SYMBOL TABLE

SNDBFR= xxxxxx G      SNDIT =  0080      STAT37= xxxxxx G      STKBLD= xxxxxx G      STPFLG=  0040
STPKEY=  0020         STRING=  0010      SYSERR= xxxxxx G      TABPTR= xxxxxx G      TAGPTR= xxxxxx G
TCOL  = xxxxxx G      TEMPX = xxxxxx G   TEMPY = xxxxxx G      TRCFLG=  0020         TRIG  = xxxxxx G
TYPARG= xxxxxx G      T51DEV=  0025      UNADR = xxxxxx G      UNDEF =  0080         VALTG = xxxxxx G
VALUND=  0040         WINERR   0116R     XAXIS = xxxxxx G      XDEFN    0582R        XLAST = xxxxxx G
XMAXW = xxxxxx G      XMINS = xxxxxx G   XMINW = xxxxxx G      XNEW  = xxxxxx G      XSF   = xxxxxx G
YDEFN    05A9R        YDEF.    05A6R     YLAST = xxxxxx G      YMAXW = xxxxxx G      YMINS = xxxxxx G
YMINW = xxxxxx G      YNEW  = xxxxxx G   YSF   = xxxxxx G      YTEST    0598R        YTST     054QR
ZDRAW = xxxxxx G      ZGIN  = xxxxxx G   ZMOVE = xxxxxx G      ZRDRAW= xxxxxx G      ZRMOVE= xxxxxx G

  ABS.    0000     00
          05E5     01
ERRORS DETECTED: 0     WARNINGS POSTED: 1     FREE CORE: 1870. WORDS
.SY:GRAF/C(DK1:SEICLI,GRAF
```

CROSS REFERENCE TABLE (CREF V01-03.)

1-276

A.END 1-196#
A.MAX 1-197#
A.PRIM 1-192#
A.PTR 1-194#
A.SEC 1-193# 1-324# 20-65 21-12*
A.STAT 1-191# 10-74 21-13*
A.STRT 1-195#
A16X 1-329# 22-30 23-33 23-37 23-57 23-61
A8X 10-15# 13-19 13-23 13-32 14-27 14-31
A8X 1-328# 10-14# 10-65 10-70 13-34 14-19 22-42
A9X 12-11 22-5# 22-19
ABRFLG 1-26#
ADDAIX 8-80#
ADDARG 8-24 8-34 8-59 8-66 8-74#
ADRDEV 10-8# 10-19
AFAIL 1-155#
AMAT 1-154# 14-22
APLOT 20-24 20-47 20-77#
ARRAY 1-83#
ASTR 1-153#
ATLOD 1-216#
ATSNTG 1-138#
ATVLD 1-204#
AXIERR 8-10# 8-21 8-22
AXISCA 7-16 7-19 8-14#
BAKSTG 1-146#
BANK 1-250#
BFRALC 10-10# 10-20
BFRSTT 1-201#
BLINK 1-222#
BMAT 1-157#
BRKCNT 1-73#
BSTR 1-156#
BUSACT 1-202#
CDOPTR 1-71#
CDSPTR 1-70#
CHAR 1-189#
CHRCNT 1-242#
CLIP 20-15 20-20 22-7# 23-89
CLIPDN 22-40 22-50 22-52#
CLIPIT 20-37 20-54 23-17#
CLPTR 1-20#
CLRARG 5-2# 5-6
CMAT 1-159#
COLCNT 1-244# 1-340# 10-44* 10-69 10-71* 11-32 12-7* 16-27 18-17 18-31 20-4 20-88
COSCOD 2-6# 2-24
COSTHA 1-288# 1-381 2-27 18-13 18-33
CRDCJ 1-181#
CREOF 1-183#
CREOI 1-182#
CREOT 1-185#
CRETX 1-184#
CRLF 16-42# 16-73
CRNORM 1-180#
CRSTAT 1-178#

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

	17-9	17-13	17-14	17-24	17-25	18-21	18-37	20-5	20-9	20-71	20-72	20-78	20-79	20-85
	20-89													
FLIN	1-330#	8-53	11-31	11-41	23-82	23-86								
FM	1-335#	3-30	3-51	8-50	11-27	11-37	18-12	18-16	18-28	18-32	23-79			
FMTVLD	1-203#													
FNFLG	1-27#													
FORTG	1-126#													
FPA	11-5#	11-9	11-34	16-43#	16-78									
FPADD	8-8#	8-64												
FPB	1-285#	11-5#	11-12	11-24	12-6	12-13								
FPC	1-285#	17-7	17-14											
FPCMP	4-10	4-23	22-4#	22-12	22-25	22-36	22-48							
FPDIV	8-8#	8-36												
FPMUL	1-326#	8-63	10-7#											
FPONE	1-381	1-392#	1-394											
FPSUB	8-8#	8-32												
FPZERO	1-367	1-392#	8-26											
FRES	1-331#													
FRET	1-330#	1-398	2-18	3-58	4-20	4-33	6-22	6-37	8-55	11-43	16-33	16-58	16-70	17-10
	17-15	17-26	18-39	20-11	20-73	20-80	20-91	23-88						
FS	1-334#	3-19	3-26	3-40	3-47	6-18	6-33	11-25	11-35	16-51	16-63	18-18	23-68	23-72
	23-77													
FSAV	1-331#	16-57	16-69											
FSWP	1-331#													
GETIT	4-8	4-21	5-3#	6-6	6-23	7-11								
GINSET	1-290#	10-28	12-4#											
GINSTK	12-5	12-12	13-6#											
GLBFLG	1-31#													
GOSTG	1-125#													
GRAF	1-275#	1-276#												
GRAFER	2-15	14-23	14-34#											
GRAFRT	14-21	14-33	14-36#											
GRAPH	9-13	9-14	9-15	9-16	9-17	10-6#	10-17#							
GRFAGN	10-49#	10-77	12-17											
GRFDR	1-283#	17-11	20-12#											
GRFDRW	10-81	20-2#												
GRFED	10-62#	11-44												
GRFERT	10-48	10-63	10-68	10-74#	11-16									
GRFGIN	10-52	11-9#												
GRFINI	1-355#	1-365#												
GRFMOV	10-82	16-18#												
GRFNRM	10-49	10-81#												
GRFREL	10-55	10-83#												
GRFSCN	10-37	10-42	14-7#											
GSTKEQ	13-12	13-33#												
GSTKEX	13-24	13-35#												
IMXTG	1-129#													
INPUTK	1-219#													
INPVAL	11-5#	11-11	11-14											
IOBFRI	1-266#													
IOCLNR	10-13#	10-78												
IOFLGS	1-246#	16-81#												
IOFINC	1-190#	10-25	10-50											
ITM1TG	1-136#													
ITM2TG	1-137#													
ITTDEV	1-260#													
JMPX	1-55#													

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

PGMTG	1-128#													
PLOSTG	1-123#													
PLOTVL	1-284#	16-34	16-47#	17-27	20-74	20-81								
PLTIT	16-59	16-71	16-77#											
PLTOK	20-62	20-65#												
PNDEOF	1-45#													
PNDFLG	1-44#													
PNTNTG	1-132#													
PNTSTG	1-130#													
POINT	1-241#	11-10*	11-13*	16-79*										
PPMODE	1-224#													
PRIDEF	1-206#													
PRIVAL	16-41#	16-82												
PRTTG	1-141#													
PSCTG	1-131#													
PSHFPN	1-327#	8-25	8-27	8-31	8-35	8-38	8-60	8-62	22-9	22-11	22-22	22-31	22-33	22-45
PSHRET	7-7#	7-10	8-14											
PULFPN	1-327#	2-23	2-28	8-16	8-29	8-40	8-67							
QDRAW	9-13#	9-13#												
QGIN	9-17#	9-17#												
QMOVE	9-15#	9-15#												
QTDRAW	9-14#	9-14#												
QRMOVE	9-16#	9-16#												
QROTAT	2-3#	2-11#												
QSCALE	1-284#	7-10#												
QVIEW	1-284#	6-5#												
QWINDO	1-284#	4-7#												
QWINEX	4-9	4-22	4-35#	4-38										
R0	1-38#	1-322#	5-5*	10-24*	10-38	10-43	13-7*	14-18*	14-28*	20-45*	20-63			
R1	1-38#	1-322#	10-40	10-45	12-4*	12-10	14-20*	14-32*	20-14*	20-19*	20-32*	20-51*	22-8	22-29
	23-31	23-55												
R10	1-38#	20-16*	20-21*	20-23	20-26	20-29	20-46							
R11	1-38#	22-7*	22-52											
R12	1-38#													
R13	1-38#													
R14	1-38#													
R15	1-38#													
R16	1-38#													
R17	1-39#													
R18	1-39#													
R19	1-39#													
R2	1-38#	1-322#	14-24*	14-29	20-34*	20-49*	23-35	23-59						
R20	1-39#	7-6#	7-15*	7-18*	8-30	8-54	8-74	12-6	12-13	13-20*	13-33*			
R21	1-39#	8-28	8-39	8-80*	8-81	8-82*	8-84*	8-85	12-8	12-15	13-17*	13-21	13-35*	
R22	1-39#													
R23	1-39#													
R3	1-38#	1-322#	23-34*	23-56*	23-78	23-81	23-83							
R4	1-38#	1-322#	23-32*	23-58*	23-67	23-71	23-87							
R5	1-38#	1-322#	23-38*	23-60*	23-76									
R6	1-38#	1-322#	23-36*	23-62*	23-73									
R7	1-38#	1-322#	23-30*	23-54*	23-69	23-85								
R8	1-38#													
R9	1-38#													
RECLFG	1-217#													
RELDRW	10-83	17-4#												
RELMOV	10-84	17-22#												
RFAIL	1-152#													

CROSS REFERENCE TABLE (CREF V01-03.)

RMAT	1-151#													
RPLOT	17-4	17-22	18-5#											
RPTCTL	1-212#													
RSTR	1-150#													
RTRN	7-7#	7-13	7-20	8-12	8-68									
RTRNTG	1-140#													
RUNFLG	1-32#													
RUNN	1-218#													
SBP	1-47#													
SCALEF	3-16#	4-34	6-38											
SCALER	1-82#	13-15												
SCL	7-5#	7-9												
SECDEF	1-205#													
SEMITG	1-144#													
SETARG	2-4#	2-12												
SINCOD	2-5#	2-19												
SINTHA	1-288#	1-373	2-22	18-15	18-29									
SNDBFR	10-11#	10-76												
SNDIT	1-228#													
STAT37	1-247#													
STKBLD	10-9#	10-29												
STPFLG	1-33#													
STPKEY	1-214#													
STRING	1-84#													
SYSERR	1-43#													
T51DEV	1-261#													
TABPTR	1-243#	1-338#	10-39*	10-64	10-66*	11-42	12-14*	16-20	18-11	18-27	20-8	20-84		
TAGPTR	1-239#													
TCOL	1-245#	1-341#	10-46*	10-72	12-9*									
TEMPX	1-289#	4-17	4-31	6-12	6-17	6-27	6-32	8-15	8-37	8-51	8-61	17-8	20-10	20-17
	20-31	20-48	20-78											
TEMPY	1-289#	4-19	4-32	17-9	20-6	20-79								
TRCFLG	1-34#													
TRIG	2-8#	2-21	2-26											
TYPARG	2-4#	5-2#	5-7											
UNADR	10-12#	10-77												
UNDEF	1-81#	13-27												
VALTG	1-134#													
VALUND	1-88#	13-29												
WINERR	4-12	4-25	4-36#											
XAXIS	1-248#													
XDEFN	23-23	23-30#												
XLAST	1-288#	1-369	3-25	3-34	16-25	17-13	17-24	18-20	20-12	20-33	20-50	20-71	20-86	
XMAXW	1-286#	1-385	3-18	4-28	22-21	23-29								
XMINS	1-287#	1-379	6-29	6-34	11-36	16-56								
XMINW	1-286#	1-375	3-20	3-32	4-14	4-30	6-9	7-17	11-40	16-52	22-10	23-22		
XNEW	1-288#	3-27	4-14	6-9	16-23	16-50	17-6	17-8	17-24	18-22	20-71	20-78		
XSF	1-287#	1-396	3-29	3-36	11-38	16-54								
YDEF	23-48	23-53#												
YDEFN	23-46	23-54#												
YLAST	1-288#	1-371	3-46	3-55	16-32	17-14	17-25	18-36	20-72	20-90				
YMAXW	1-286#	1-390	3-39	4-31	22-44	23-53								
YMINS	1-287#	1-380	6-14	6-19	11-26	16-68								
YMINW	1-286#	1-377	3-41	3-53	4-15	4-32	6-10	7-14	11-30	16-64	22-32	23-45		
YNEW	1-289#	3-48	4-15	6-10	16-30	16-62	17-7	17-9	17-25	18-38	20-72	20-79		
YSF	1-287#	1-397	3-50	3-57	11-28	16-66								
YTEST	23-25	23-40#												

CROSS REFERENCE TABLE (CREF V01-03.)

YTST	22-17	22-27	22-29#
ZDRAW	9-13	9-13#	
ZGIN	9-17	9-17#	
ZMOVE	9-15	9-15#	
ZRDRAW	9-14	9-14#	
ZRMOVE	9-16	9-16#	

CROSS REFERENCE TABLE (CREF V01-03.)

ENTRY	9-5#	9-13	9-14	9-15	9-16	9-17								
Q	1-356#	1-369	1-371	1-373	1-375	1-377	1-379	1-380	1-385	1-386	1-390	1-391	1-396	1-397
R	1-361#	1-368	1-370	1-372	1-374	1-376	1-378	1-384	1-389	1-395				
SAVRTN	1-342#	2-11	4-7	6-5	13-6	14-7								
SE1	1-3#	13-26												
SET	1-349#	1-381	4-14	4-15	4-31	4-32	6-9	6-10	17-6	17-7	17-8	17-9	17-13	17-14
	17-24	17-25	20-71	20-72	20-78	20-79								

CROSS REFERENCE TABLE (CREF V01-03.)

W 13-29

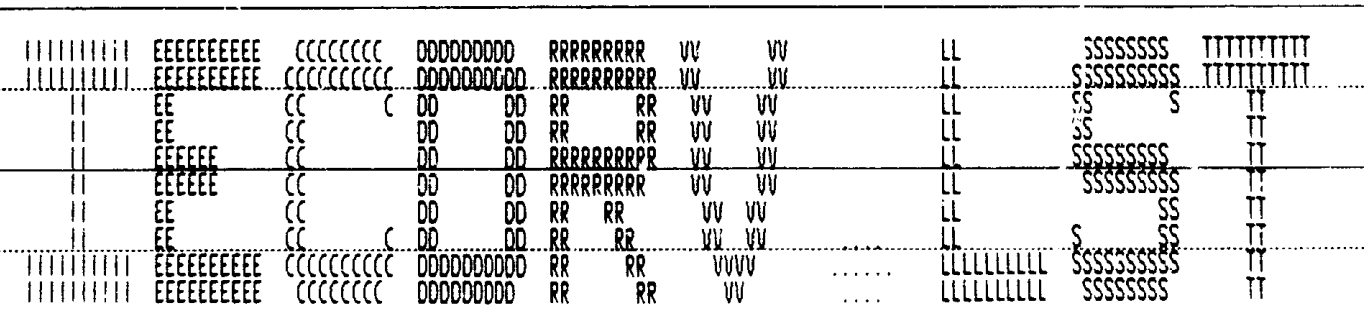


TABLE OF CONTENTS

```
                        .TITLE  IECDRV   IEC INTERFACE SERVICE ROUTINE
                        .IDENT  /SBK035/
                        .GLOBL  INTACP  ; INITIALIZE IEC HARDWARE AS ACCEPTOR
                        .GLOBL  IECOFF  ; INITIALIZE ACCEPTOR AND UNLOAD THE BUS
                                        ; AS WELL AS DISABLE ATN
                        .GLOBL  IECRD   ; READ A BYTE FROM THE IEC BUS
                        .GLOBL  IECSND  ; WRITE A BYTE TO THE IEC BUS
                        .GLOBL  INTSRC  ; INITIALIZE IEC HARDWARE AS SOURCE
                        .GLOBL  IECIFC  ; SEND "IFC" ON IEC BUS
                        .GLOBL  ATNON   ; TURN ON ATENTION
                        .GLOBL  ATNOFF  ; TURN OFF ATENTION
                        .GLOBL  PIASRQ  ; THE SRQ HALF OF THE IEC PIA
                        .GLOBL  PIAEOI  ; THE EOI HALF OF THE IEC PIA
                        .GLOBL  IECOUT  ; OUTPUT A BUFFER TO THE IEC BUS
                        .GLOBL  EOION   ; SET EOI
                        .GLOBL  EOIOFF  ; RESET EOI
                        .GLOBL  CHAR    ; I/O CHARACTER
                        .GLOBL  ETXCHR,EOLCHR,NULCHR    ; SPECIAL DELIMITER CHARACTERS
                        .GLOBL  PIALT
                        .GLOBL  STKBLD
                        .GLOBL  A9X
                        .GLOBL  A5X
                        .GLOBL  FIX1
                        .GLOBL  FLOAT1
                        .GLOBL  PULFPN
                        .GLOBL  LOCLNR

                ;       xxxxxxxx SEE SIOVAL FOR GEORGE'S CHANGES xxxx
```

```
                                        .SBTTL  IECDRV--IEC BUS HARDWARE DRIVERS
                    0001                IECDRV=1
                                        .GLOBL  IECDRV
                                ;
                                ;
                                ;INTACP   SUBROUTINE
                                ;
                                ;
                                ;THIS SUBROUTINE INITIALIZES THE PIA AS AN ACCEPTOR.
                                ;THIS IS CALLED FIRST WHEN THE CALCULATOR RECEIVES DATA
                                ;AND MESSAGES.
                                ;
                                ;
0000    07                      INTACP: TPA
0001    0F                              .BYTE   HOF             ; THAT'S A SEI
0002    CE    0000G                     LDX     PIASRQ,I        ; SORRY ABOUT THE READABILITY
0005    E6    00                        LDA B   0,X
0007    C4    6C                        AND B   H6C,I   ; SET NRFD & NDAC =0
                                                              ; ALSO IFC & EOI =1
0009    E7    00                        STA B   0,X
000B    C6    30                        LDA B   60,I    ;ADDRESS DATA DIRECTION REG. A
000D    F7    0001G                     STA B   PIAEOI+1        ; "
0010    E7    01                        STA B   1,X             ;ADDRESS DDR B
0012    C6    9F                        LDA B   237,I   ;CHANGE DDR B
0014    E7    00                        STA B   0,X             ; "
0016    7F    0000G                     CLR     PIAEOI  ;CHANGE DDR A
0019    C6    37                        LDA B   67,I    ;ADDRESS PIAEOI
001B    F7    0001G                     STA B   PIAEOI+1        ; "
001E    E7    01                        STA B   1,X             ;ADDRESS PIASRQ AND SET TALK=0
0020    06                              TAP
0021    39                              RTS
                                ;
0022    84    7F        EXTRA:  AND A   -1-128.,I       ; TURN OFF NDAC
0024    B7    0000G             STA A   PIASRQ
0027    39                      RTS                     ; RETURN
                                ;
                                ;
                                ;'IECOFF'
                                ;
                                ;THIS ROUTINE INITIALIZES THE CALC. AS A LISTENER BUT
                                ;GETS COMPLETELY OFF THE BUS SO IT DOESN'T HAVE TO HAND SHAKE.
                                ;IT ALSO CLEARS ATENTION WHEN IT FINISHES
                                ;
0028    8D    D6        IECOFF: BSR     INTACP
002A    C6    0C                LDA B   H0C,I   ; SET ALL LINES HIGH EXCEPT REN,NDAC,NRFD
002C    F7    0000G             STA B   PIASRQ
002F    7E    00FC'             JMP     SRQRDY          ; MAKE SURE NO SRQ HAPPEND WHILE GONE
                                .BLKB   1
                                ;
                                ;
                                ;'IECRD'   SUBROUTINE
                                ;
                                ;ACCEPTOR HANDSHAKE FUNCTION
                                ;THIS SUBROUTINE PROVIDES THE CALCULATOR TO
                                ;GUARANTEE RECEPTION OF VARIOUS MESSAGES.
                                ;
```

```
                                ;'INTACP' SUBROUTINE MUST BE CALLED
                                ;BEFORE ENTRY TO THIS ROUTINE
                                ;
                                ;EXIT PARAMETER:  CHAR HOLDS THE BYTE
                                ;                 READ FROM DIO BUS
                                ;
                                
0033  B6  0000G    IECRD:  LDA A  PIASRQ          ; SET NRFD = 1
0036  8A  10                 ORA A  16.,I
0038  B7  0000G              STA A  PIASRQ
003B  F6  0000G    1$:     LDA B  PIASRQ          ; READ DAV
003E  C5  40                 BIT B  64.,I         ; WAIT FOR IT
0040  27  F9                 BEQ    1$
0042  C4  FF                 AND B  357,I         ; SET NRFD = 0
0044  F7  0000G              STA B  PIASRQ
0047  37                     PSH B                ; SAVE FOR A WHILE
0048  BD  00E1'              JSR    IECEOI        ; LOOK FOR AN EOI
004B  97  00G                STA A  CRSTAT,D      ; SAVE EOI STATUS
004D  D7  00G                STA B  CHAR,D        ; ALSO GRABS CHARACTER
004F  32                     PUL A                ; GET BACK STATUS
0050  8A  80                 ORA A  128.,I        ; SET NDAC=1
0052  B7  0000G              STA A  PIASRQ
0055  B6  0000G    2$:     LDA A  PIASRQ
0058  85  40                 BIT A  64.,I         ; WAIT FOR DAV TO GO AWAY
005A  26  F9                 BNE    2$
005C  20  C4                 BRA    EXTRA                ; GO FOR CODE
                                ;
                                ;
                                ;'IECSND'  SUBROUTINE
                                ;
                                ;
                                ;THIS SUBROUTINE PROVIDES
                                ;TRANSMISSION OF DATA
                                ;ON DIO BUS TO ACCEPTORS.
                                ;ENTRY PARAMETERS:
                                ;       ACC B IS THE BYTE TO WRITE
                                ;       ON IEC-DIO BUS
                                ;       DAV:LOW(AT PIA)
                                ;
                                ;ALL THE LOGIC ARE OPPOSITE
                                ;OF SIGNALS ON IEC-BUS.
                                ;
005E  86  34       IECSND: LDA A   H34,I  ; SET DAV HIGH
0060  B7  0001G              STA A  PIAEOI+1        ; DAV<---HIGH
0063  B6  0000G              LDA A  PIASRQ  ;READ NRFD AND NDAC
0066  84  90                 AND A  220,I
0068  27  1A                 BEQ    IOERROR ;GO TO IOERROR
                                                    ;IF NRFD AND NDAC=0
006A  F7  0000G              STA B  PIAEOI  ;OUTPUT DIO BY (WBYTE)
006D  B6  0000G    1$:     LDA A  PIASRQ  ;READ NRFD
0070  84  10                 AND A  20,I
0072  26  F9                 BNE    1$      ;NRFD=0?
0074  86  3C                 LDA A   H3C,I  ; SET DAV LOW
0076  B7  0001G              STA A  PIAEOI+1        ;
0079  F6  0000G    2$:     LDA B  PIASRQ  ;READ NDAC
007C  2B  FB                 BMI    2$
007E  86  34                 LDA A   H34,I  ; SET DAV HIGH
```

```
                                                STA A   PIAE01+1
                                                RTS
                        IOERROR: LDA A   ERIOE,I ; SET IOE ERROR
                                 STA A   ERRCD,D
                                 RTS
                                ;
                                ;
                                ;'INTSRC'  SUBROUTINE
                                ;
                                ;
                                ;THIS SUBROUTINE INITIALIZES THE
                                ;PIA AS A SOURCE FUNCTION
                                ;
                        INTSRC: TPA
                                .BYTE   HOF             ; THAT'S SET THE HARD WAY
                                LDA B   60,I
                                STA B   PIAE01+1        ;ADDRESS TO DATA
                                                        ; DIRECTION REGISTERS A&B
                                STA B   PIASRQ+1        ; "
                                LDA B   17,I    ;CHANGE DDRB
                                STA B   PIASRQ  ;"
                                LDA B   377,I   ;CHANGE DDRA
                                STA B   PIAE01  ;"
                                LDA B   64,I    ;CHOOSE PIAE01
                                STA B   PIAE01+1        ;"
                                LDA B   77,I    ;CHOOSE PIASRQ, AND SET TALK=1
                                STA B   PIASRQ+1        ;"
                                TAP
                                BRA     E01OFF          ; MAKE SURE E01 IS OFF
                                ;
                                ;
                                ;'IECIFC'  SUBROUTINE
                                ;
                                ;THIS ROUTINE OUTPUTS IFC PULSE FOR >100 US
                                ;
                        IECIFC: JSR     IECOFF          ; GET OFF BUS ALSO ***
                                                        ; THIS IS PATCH FOR INIT AS WELL
                                ORA B   2,I
                                STA B   PIASRQ
                                LDA A   24.,I
                        1$:     DEC A
                                BNE     1$
                                AND B   375,I
                                STA B   PIASRQ
                                RTS
                                ;
                                ;
                                ;'E01ON'        SUBROUTINE
                                ;
                                ;
]]]]]
                        E01ON:  LDA A   PIASRQ          ; GET DATA
                                ORA A   1,I             ; SET E01
                                BRA     X2
                                .BLKB 2
                                ;
                                ;
                                ;'E01OFF'       SUBROUTINE
```

```
                                        ;
                                        ;
00C6  B6  0000G   EOIOFF: LDA A  PIASRQ         ; READ DATA REG.
00C9  84  FE              AND A  #0FE,I ; RESET EOI
00CB  B7  0000G   X2:     STA A  PIASRQ         ; SAVE DATA
00CE  39                  RTS
                                        ;
                                        ;
                          ;'ATNON'  SUBROUTINE
                                        ;
                                        ;
00CF  B6  0000G   ATNON:  LDA A  PIASRQ         ; LOOK AT DAV FIRST
00D2  85  40              BIT A  #64.,I         ; IN ORDER TO TAKE CONTROL SYNCRONOUSLY (TCS)
00D4  26  F9              BNE    ATNON
00D6  84  F7              AND A  #367,I         ; THEN ASSERT ATTENTION
00D8  20  F1              BRA    X2             ; EXIT
                                        ;
                                        ;
                          ;'ATNOFF'  SUBROUTINE
                                        ;
                                        ;
00DA  B6  0000G   ATNOFF: LDA A  PIASRQ
00DD  8A  08              ORA A  #10.I
00DF  20  EA              BRA    X2
```

```
                                        .SBTTL  EOIDRV--EOI DRIVERS
                                        .GLOBL  IECEOI
                                        .GLOBL  ONTBL
                                        .GLOBL  PIALT           ; DC LEVEL OF EOI IS HERE
                                ;
                                ; THIS ROUTINE CHECKS THE LEVEL OF EOI AND IF TRUE
                                ; THEN SETS THE EOI BIT IN PNDFLG FOR THE EOI ON UNIT
                                ; IT ALSO RETURNS WITH THE FOLLOWING:
                                ;       ACC A -- 0 IF NO EOI, CREOI (4) IF HAD EOI
                                ;       ACC B -- THE CHARACTER CONTAINED IN PIAEOI
                                ;                WHICH MY BE OF USE IF THIS ROUTINE CALL FROM IECRD
                                ;
00E1    4F              IECEOI: CLR A
00E2    F6      0000G           LDA B   PIALT           ; LOOK AT EOI --- DC LEVEL
00E5    C5      10              BIT B   16.,I
00E7    27      08              BEQ     EOIRDY          ; NOPE
00E9    D6      00G             LDA B   PNDFLG,D        ; SET EOI INTERUPT BIT
00EB    CA      08              ORA B   10,I
00ED    D7      00G             STA B   PNDFLG,D
00EF    86      04              LDA A   4,I             ; SET CREOI ALSO
00F1    F6      0000G   EOIRDY: LDA B   PIAEOI          ; CLEAR INTERUPT AND READ CHAR
00F4    39                      RTS
                                .BLKB   Z
```

```
                                        .SBTTL  SRQDRV--SRQ DRIVERS
                                        .GLOBL  SRQRDY
                                ;THESE ROUTINES HANDLE THE SERVICE REQUEST INTERUPTS
                                ;LATER CODE NEEDS TO BE ADDED TO HANDLE DEVICE 0 SPECIALLY
                                ;
00FC    86      0000G   SRQRDY: LDA A   PIASRQ          ; SRQ CLEAR INTERRUPT
00FF    85      20              BIT A   32.,I           ; TEST IF SRQ IS STILL DOWN
0101    27      0C              BEQ     SRQDID
0103    96      00G             LDA A   GLBFLG,D        ; IS SRQ DISABLED
0105    85      08              BIT A   DISSRQ,I
0107    26      06              BNE     SRQDID
0109    96      00G             LDA A   PNDFLG,D        ; SET SRQ INTERUPT
0108    8A      10              ORA A   20,I
010D    97      00G             STA A   PNDFLG,D
010F    39              SRQDID: RTS
```

```
                                        .SBTTL  IECOUT--OUTPUT DATA BUFFER TO IEC BUS
                                    ;THIS ROUTINE TRANSFERS THE INFORMATION IN THE OUTPUT BUFFER
                                    ;POINTED TO BY THE CHAN. A POINTERS.
                                    ;
0110  96  00G     IECOUT: LDA A   A.STAT,D        ; IS DEVICE ACTIVE?
0112  85  10              BIT A   BUSACT,I
0114  26  24              BNE     IECOK           ; BUS IS READY
0116  8A  10              ORA A   BUSACT,I        ; MAKE BUS ACTIVE
0118  97  00G             STA A   A.STAT,D
011A  BD  00CF'           JSR     ATNON
011D  BD  0089'           JSR     INISRC          ; NEED TO ADDRESS DEVICE
0120  D6  00G             LDA B   A.PRIM,D        ; GET THE PRIMARY ADDRESS
0122  CA  20              ORA B    H20,I  ; MAKE IT IEC
0124  BD  005E'           JSR     IECSND
0127  D6  00G             LDA B   A.SEC,D         ; GET SECONDARY ADDRESS
0129  C1  20              CMP B   32.,I           ; IF 32. THEN NO SECONDARY ADDRESS
012B  27  05              BEQ     IECSTP
012D  CA  60              ORA B    H60,I  ; MAKE IT IEC
012F  BD  005E'           JSR     IECSND
0132  BD  00DA'   IECSTP: JSR     ATNOFF
0135  96  00G             LDA A   ERRCD,D         ; ANY ERRORS?
0137  27  01              BEQ     IECOK
0139  39                  RTS
013A  DE  00G     IECOK:  LDX     A.PTR,D         ; GET POINTER TO CHARACTER
013C  9C  00G     IECGO1: CPX     A.END,D         ; NEED EOI WITH THIS CHAR?
013E  26  00              BNE     IECGO
0140  96  00G             LDA A   NOOUT,D         ; EOI GOES ON LAST CHAR?
0142  85  80              BIT A   SNDIT,I
0144  27  05              BEQ     IECGO           ; NOPE
0146  BD  008D'           JSR     EOION           ; SET EOI
0149  26  00              BNE     IECGO
014B  E6  00      IECGO:  LDA B   0,X              ; GET A CHARACTER
014D  BD  005E'           JSR     IECSND          ; SEND IT
0150  96  00G             LDA A   ERRCD,D         ; I/O ERROR?
0152  26  07              BNE     IECCLN
0154  9C  00G             CPX     A.END,D         ; DONE?
0156  27  03              BEQ     IECCLN          ; GO CLEAN UP
0158  08                  INX
0159  20  E1              BRA     IECGO1          ; GO OUTPUT NEXT CHAR.
015B  BD  00C6'   IECCLN: JSR     EOIOFF          ; RESET EOI
015E  39          IECEXT: RTS                     ; RETURN
```

```
                                        .SBTTL  IECIN---INPUT FROM IEC BUS
                                        .GLOBL  IECIN
                                        .GLOBL  SCRMS,XAXIS
                                ;THIS ROUTINE INPUTS ASCII INFORMATION FROM THE IEC BUS
                                ;IF AN EOI IS DETECTED THEN APPROPRIATE STATUS IS SET
                                ;       1) CREOI IS SET
                                ;       2) BUSACT IS RESET
                                ;       3) WHO KNOWS WHAT ELSE
 015F  96  000  IECIN:  LDA A   A.STAT,D        ; IS DEVICE ACTIVE
 0161  85  10           BIT A   BUSACT,I
 0163  26  29           BNE     JECOK           ; BUS IS READY
 0165  8A  10           ORA A   BUSACT,I        ; MARK BUS ACTIVE
 0167  97  000          STA A   A.STAT,D
 0169  BD  000E'        JSR     ATNON
 016C  BD  0089'        JSR     INTSRC          ; NEED TO ADDRESS DEVICE
 016F  D6  000          LDA B   A.PRIM,D        ; GET PRIMARY ADDRESS
 0171  CA  40           ORA B    H40,I  ; MAKE IT IEC
 0173  BD  005E'        JSR     IECSND          ; SEND IT
 0176  D6  000          LDA B   A.SEC,D         ; GET SECONDARY
 0178  C1  20           CMP B   32.,I           ; IF 32 THEN NO SEC. ADDRESS
 017A  27  06           BEQ     JECSTP
 017C  CA  60           ORA B    H60,I  ; MAKE IT IEC
 017E  BD  005E         JSR     IECSND          ; SEND IT
 0181  BD  0000'  JECSTP: JSR   INTACP          ; SET UP TO BE ACCEPTOR
 0184  86  0C           LDA A    H0C,I  ; TURN OFF ATENTION AND
 0186  B7  0000G        STA A   PIASRA          ; SET NRFDA,NDAC TO 0
                        ;xxxxxxxxxxxxxxx QUESTION ABOUT TIMING HERE
 0189  96  000          LDA A   ERRCD,D         ; ANY ERRORS
 018B  27  01           BEQ     JECOK
 018D  39               RTS
 018E  DE  000  JECOK:  LDX     A.PTR,D         ; POINTER TO BUFFER
 0190  BD  0033' JECXER: JSR    IECRD           ; GET CHAR
 0193  96  000          LDA A   CHAR,D          ; GET CHARACTER
 0195  D6  000          LDA B   IOFUNC,D        ; IF READ SKIP
 0197  C1  0E           CMP B   14.,I
 0199  27  1C           BEQ     4$
 019B  91  000          CMP A   ETXCHR,D        ; TEST FOR ETX
 019D  26  07           BNE     3$
 019F  C6  10           LDA B   CRETX,I         ; SET ETX
 01A1  D7  000          STA B   CRSTAT,D
 01A3  09               DEX
 01A4  20  25           BRA     JECEXT
 01A6  BD  0000G  3$:   JSR     SCRMS
 01A9  D6  000          LDA B   NULCHR,D        ; NULL CHARACTER
 01AB  2B  06           BMI     1$
 01AD  11               CBA
 01AE  26  03           BNE     1$
 01B0  09               DEX
 01B1  20  06           BRA     2$
 01B3  91  000  1$:     CMP A   EOLCHR,D        ; EOL?
 01B5  27  00           BEQ     JECRE.
 01B7  A7  00   4$:     STA A   0,X
 01B9  D6  000  2$:     LDA B   CRSTAT,D        ; EOI?
 01BB  26  0E           BNE     JECEXT
 01BD  9C  000          CPX     A.MAX,D         ; FULL BUFFER
 01BF  27  0A           BEQ     JECEXT
 01C1  08               INX                     ; MOVE POINTER
```

IECIN---INPUT FROM IEC BUS

```
    58  01C2  20  CC          BRA   JECXFR      ; GO GET NEXT CHARACTER
    59  01C4  09        JECRE: DEX
    60  01C5  D6  00G         LDA B CRSTAT.D
    61  01C7  CA  01          ORA B CRNORM,I    ; MARK "CR" FOUND"
    62  01C9  D7  00G         STA B CRSTAT.D
    63  01CB  08        JECEXT: INX             ; MOVE POINTER TO PROPER PLACE
    64  01CC  DF  00G         STX   A.END.D     ; SAVE POINTER TO LAST CHAR
    65  01CE  39              RTS
```

```
                                        .SBTTL  POLL----POLL DRIVER
                                ;THIS ROUTINE PREFORMS THE POLL FUNCTION ON THE IEC BUS
                                ;DEVICE ZERO NEEDS TO BE DISALLOWED.
                                ;
                                        .GLOBL  LOCTG
                                        .GLOBL  POL!
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;THIS ROUTINE EXTRACTS AN INTEGER NUMBER OFF THE STACK
                                ;       AND IS SPECIALLY BUILT FOR THE POLL COMMAND
                                ;
01CF    32                      POLVAL: PUL A                   ; SAVE RETURN ADDRESS
01D0    97      01G                     STA A   DREXTA+1,D
01D2    32                              PUL A
01D3    97      02G                     STA A   DREXTA+2,D
01D5    32                              PUL A
01D6    32                              PUL A                   ; VALID INPUT VARIABLE
01D7    40                              TST A
01D8    26      11                      BNE     POLER.
01DA    30                              TSX
01DB    EE      00                      LDX     0,X             ; GET POINTER TO VALUE
01DD    08                              INX
01DE    BD      0000G                   JSR     FIX1            ; FIX IT
01E1    26      08                      BNE     POLER.
01E3    E6      03                      LDA B   FIX1A,X         ; GET MSBYTE
01E5    26      04                      BNE     POLER.
01E7    E6      04                      LDA B   FIX1B,X         ; GET LSBYTE
01E9    20      02                      BRA     POLEX.
01EB    C6      80      POLER.: LDA B   200,I           ; ERROR VALUE
01ED    31              POLEX.: INS                     ; FINSH CLEARING STACK
01EE    31                      INS
01EF    7E      0000G           JMP     DREXTA
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;       THIS IS MAIN ENTRY POINT FOR POLL
                                ;
01F2    B6      0000G   POLL:  LDA A   PIALT           ; TURN ON I/O LIGHT
01F5    8A      40              ORA A   100,I
01F7    B7      0000G           STA A   PIALT
01FA    30                      TSX                     ; SET UP STACK
01FB    08                      INX
01FC    DF      00G             STX     R0,D
01FE    DE      00G             STX     R11,D           ; MARK TOP OF STACK
0200    86      00G             LDA A   EOLTG,I
0202    36                      PSH A
0203    BD      0000G           JSR     LOCTG
0206    D6      00G             LDA B   R0,D            ; FIND REAL EOL TAG
0208    96      01G             LDA A   R0+1,D          ; BACK UP TO RESULT AREAS
020A    80      12              SUB A   18.,I
020C    97      01G             STA A   TCOL+1,D        ; MAKE POINTER TO RESULT B
020E    C2      00              SBC B   0,I
0210    D7      00G             STA B   TCOL,D
0212    DE      00G             LDX     TCOL,D          ; MAKE POINTER TO RESULT A
0214    BD      0000G           JSR     A9X
0217    DF      00G             STX     COLCNT,D        ; POINTER TO A
0219    DE      00G             LDX     TCOL,D
021B    86      00G             LDA A   EOLTG,I         ; PUT AN EOL TAG OVER THE SEMI TAG
021D    A7      00              STA A   0,X
021F    DE      00G             LDX     R11,D
```

```
                                                STX     R0,D            ; FORM PROPER STACK
                                                JSR     STKBLD          ; BUILD PROPER STACK
                                                CLR A                   ; INITIALIZE RESULTS
                                                STA A   CHAR,D
                                                STA A   R1,D            ; POLL POSITION COUNTER
                                                JSR     SPOLON          ; SEND SERIAL POLL ENABLE
                                                LDA A   ERRCD,D         ; TEST FOR ERRORS FROM ABOVE
                                                BNE     POLEXT
                                POLOOP:         BSR     POLVAL          ; GET PRIMARY
                                                CMP B   31.,I           ; TEST IF VALID PRIMARY
                                                BHI     POLERR
                                                ORA B   100,I           ; TALK ADDRESS
                                                JSR     IECSND          ; SEND IT
                                                TSX                     ; LOOK AT IAG
                                                LDA A   0,X
                                                CMP A   EOLTG,I         ; DONE?
                                                BEQ     POLIT
                                                CMP A   SEMITG,I        ; ":"
                                                BEQ     POLIT
                                                BSR     POLVAL          ; GET SECONDARY
                                                CMP B   32.,I
                                                BHI     POLERR
                                                ORA B   140,I           ; MAKE IT SECONDARY
                                                JSR     IECSND          ; SEND IT
                                POLIT:          INC     R1              ; INC. COUNTER
                                                LDA A   ERRCD,D         ; TEST FOR I/O ERRORS
                                                BNE     POLERX          ; ERROR EXIT
                                                JSR     INTACP          ; ALLOW RESPONSE
                                                JSR     ATNOFF
                                                JSR     IECRD           ; GET RESPONSE OFF OF THE BUS
                                                JSR     ATNON           ; SET UP SOURCE AGAIN
                                                JSR     INTSRC
                                                LDA A   CHAR,D
                                                BIT A   100,I           ; AFFIRMATIVE?
                                                BNE     POLEXT          ; FOUND IT
                                                PUL A                   ; WHERE FROM HERE?
                                                CMP A   EOLTG,I         ; DONE?
                                                BNE     POLOOP
                                                PSH A                   ; KEEP IT CLEAN
                                                BRA     POLERX
                                POLERR:         LDA A   ERRTSN,I
                                                STA A   ERRCD,D
                                POLERX:         CLR A
                                                STA A   R1,D            ; CLEAR RESULTS
                                                STA A   CHAR,D
                                POLEXT:
                                                LDA B   137,I           ; SEND UNTALK
                                                JSR     IECSND
                                                LDA B   31,I            ; SERIAL POLL DISABLE
                                                JSR     IECSND
                                                JSR     IECOFF          ; GET OFF THE BUS
                                                LDX     COLCNT,D        ; GET POINTER TO RESULT A
                                                LDA B   R1,D            ; GET LOCATION #
                                                BSR     STOVAL          ; STORE IN RESULT A
                                                LDX     TCOL,D          ; GET POINTER TO RESULT B
                                                LDA B   CHAR,D          ; GET RESULT STATUS
                                                BSR     STOVAL
```

```
                                                STS     R0,D            ; THIS IS A MESS TO CLEAN UP
                                                LDA A   EOLTG,I
                                                JSR     LOCTG
                                                LDX     R0,D
                                                INX
                                                INX
                                                INX
                                                STX     R0,D
                                                LDA A   EOLTG,I
                                                JSR     LOCTG           ; GET OLD SEMI TAG
                                                LDX     R0,D
                                                LDA A   SEMITG,I        ; REPLACE IT WITH SEMI TAG
                                                STA A   1,X
                                                STS     R0,D            ; RESET TO CLEAN STACK
                                                LDA A   PIALT           ; TURN OFF I/O LIGHT
                                                AND A   277,I
                                                STA A   PIALT
                                                JSR     IOCLNR          ; AND RETURN
                                        ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                        ;THIS ROUTINE FORMS AN INTEGER INTO A VALID SCALER RECIEVING
                                        ; VARIABLE
                                        ;INPUT:
                                        ;       B       IS INTEGER VALUE TO BE STORED
                                        ;       X       IS POINTER-1 TO STACKED POINTER TO NAME TABELE
                                        ;
02BF    FE      02              STOVAL: LDX     2,X             ; GET POINTER TO NAMETABLE

                                        ; xxxxx CHANGES START HERE
                                        ;       I CHANGED THE SEI/CLI TO A TPA/TAP WITH PSH/PUL

                                                TPA
                                                PSH A
02C3                                            SEI                     ;NOW TURN THEM OFF
02C3    01      0F                      .BYTE   01,17

                                                LDA A   NTATTR,X        ; GET ATTRIBUTE
                                                BIT A   SCALER,I
                                                BEQ     PERR
                                                AND A   -1-UNDEF,I      ; MARK IT DEFINED
                                                STA A   NTATTR,X
                                                JSR     A5X             ; GET POINTER TO VALUE
                                                STX     R0,D            ; SAVE IT A WHILE
                                                PSH B                   ; PUT INTEGER ON STACK
                                                CLR A
                                                PSH B
                                                PSH B                   ; HOW'S THAT FOR A TAG
                                                JSR     FLOAT1          ; FLOAT IT
                                                LDX     R0,D            ; SAVE IT
                                                JSR     PULFPN

                                                PUL A                   ;GET STATUS
                                                TAP                     ;RESTORE THE INTERRUPTS

                                        ;       xxxx CHANGES STOP HERE

                                                RTS
```

```
  171  02E3  86  000G      PERR:  LDA A  ERDOMN,I
  172  02E5  97  000G             STA A  ERRCD,D
  173  02E7  39                   RTS
  174                     ;*************************************************************
  175                     ;       ROUTINE TO INTITIALIZE POLLING
  176                     ;
  177                            .GLOBL  SPOLON          ; SERIAL POLL ON
  178  02E8  BD  0089'    SPOLON: JSR     INITSRC         ; ENABLE AS SOURCE
  179  02EB  BD  00CF'            JSR     ATNON           ; ENABLE ATN
  180  02EE  C6  3F              LDA B   77,I            ; UNLISTEN
  181  02F0  BD  005E'            JSR     IECSND          ; SEN IT
  182  02F3  C6  18              LDA B   30,I            ; SERIAL POLL ENABLE
  183  02F5  BD  005E'            JSR     IECSND          ; SEND IT
  184  02F8  39                   RTS
  185       0001'                .END
```

```
IECDRV   IEC INTERFACE SERVICE        RT-11 MMAC  VM02-10   14-OCT-76 01:34:09 PAGE 7+
SYMBOL TABLE

ABRFLG=  0040        AFAIL =  0030        AMAT  =  0010        ARRAY =  0020        ASTR  =  0020
ATLOD =  0008        ATNOFF   000ARG      ATNON    00CERG      ATSNTG= xxxxxx G     ATVLD =  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     A5X   = xxxxxx G     A9X   = xxxxxx G     BAKSTG= xxxxxx G
BANK  = xxxxxx G     BFRSTI=  0020        BLINK = xxxxxx G     BMAT  =  0004        BRKCNT= xxxxxx G
BSTR  =  0008        BUSACT=  0010        CDOPTR= xxxxxx G     COSPTR= xxxxxx G     CHAR  = xxxxxx G
CHRCNT= xxxxxx G     CLPTR = xxxxxx G     CMAT  =  0001        COLCNT= xxxxxx G     CRDC3 =  0002
CREOE =  0008        CREOI =  0004        CREOT =  0020        CRETX =  0010        CRNORM=  0001
CRSTAT= xxxxxx G     CRVLD =  0080        CSTR  =  0002        CTKN  = xxxxxx G     CURSOR= xxxxxx G
DATDEV=  0022        DIMFLG=  0004        DIRCT =  0080        DISCNT= xxxxxx G     DISSRQ=  0008
DL    = xxxxxx G     DP    = xxxxxx G     DREXTA= xxxxxx G     DREXTB= xxxxxx G     DSPDEV=  0020
DT    = xxxxxx G     EDTBFR= xxxxxx G     ENDKEY=  0040        EOFTYP=  0038        EOIOFF   00C6RG
EOION    00BDRG      EOIRDY   00F1R       EOLCHR= xxxxxx G     EOLTG = xxxxxx G     EOSTG = xxxxxx G
ERATSN= xxxxxx G     ERDONN= xxxxxx G     EREOM = xxxxxx G     ERFBER= 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     ETXCHR= xxxxxx G     EXTFLG=  0080        EXTRA    0022R
FILDEV=  0000        FIX1  = xxxxxx G     FIX1A =  0003        FIX1B =  0004        FLOAT1= xxxxxx G
FMTVLD=  0008        FNFLG =  0010        FORTG = xxxxxx G     GLBFLG= xxxxxx G     GOSTG = xxxxxx G
IECCLN   015BR       IECDRV=  0001 G      IECEOI   00E1RG      IECEXT   015ER       IECGO    014BR
IECGO1   013CR       IECIEC   0098RG      IECIN    015ERG      IECOFF   0028RG      IECOK    0138R
IECOUT   0110RG      IECRD    0033RG      IECSND   005ERG      IECSTP   0132R       INXTG = xxxxxx G
INPUTK=  0001        INTACP   0000RG      INTSRC   0089RG      IOBFR1= xxxxxx G     IOCLNR= xxxxxx G
IOERQR   0084P       IOFLGS= xxxxxx G     IOFUNC= xxxxxx G     ITM1TG= xxxxxx G     ITM2TG= xxxxxx G
ITTDEV=  0024        JECEXT   01CBR       JECOK    018ER       JECRE.   01C4R       JECSTP   0181R
JECXFR   0190R       JMPX  = xxxxxx G     KBDEV =  001F        KBFLAG= xxxxxx G     KBIN  = xxxxxx G
KEYFLG=  0010        KEYSTK= xxxxxx G     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     MTPDEV=  0021        MTPO2 =  0023
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     OBJATR=  0002
]]]]]
OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002        ONTBL = xxxxxx G
OPRADR= xxxxxx G     PAETG = xxxxxx G     PARM  =  0008        PERR     02E3R       PGMATR=  0002
PGMBP =  0005        PGMCD =  0009        PGMFP =  0003        PGMLEN=  0000        PGMLNN=  0007
PGMPTR= xxxxxx G     PGMTG = xxxxxx G     PIAEOI= xxxxxx G     PIALT = xxxxxx G     PIASRQ= xxxxxx G
PLOSTG= xxxxxx G     PNDEOF= xxxxxx G     PNDFLG= xxxxxx G     PNTNTG= xxxxxx G     PNTSTG= xxxxxx G
POINT = xxxxxx G     POLERR   0277R       POLERX   027BR       POLER.   01EBR       POLEXT   0280R
POLEX.   01EDR       POLIT    0253R       POLL     01F2RG      POLOOP   0232R       POLVAL   01CFR
PPMODE= xxxxxx G     PRIDEF=  0001        PRTIG = xxxxxx G     PSCTG = xxxxxx G     PULFPN= xxxxxx G
RECLFG=  0004        RFAIL =  00C0        RMAT  =  0040        RPTCTL=  0080        RSTR  =  C080
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     S8P   = xxxxxx G     SCALER=  0040        SCRMS = xxxxxx G
SECDEF=  0002        SEMITG= xxxxxx G     SNDIT =  0080        SPOLON   02E8RG      SRQDID   010FR
SRQRDY   00FCRG      STAT37= xxxxxx G     STKBLD= xxxxxx G     STOVAL   028FR       STPFLG=  0040
STPKEY=  0020        STRING=  0010        SYSERR= xxxxxx G     TABPTR= xxxxxx G     TAGPTR= xxxxxx G
TCOL  = xxxxxx G     TRCFLG=  0020        T51DEV=  0025        UNDEF =  0080        VALTG = xxxxxx G
VALUND=  0040        XAXIS = xxxxxx G     X2       00C8R
. ABS.   0000      00
         02F9      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2406  WORDS
.SY:IECDRV/C/OK1:SEICLI,IECDRV
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

PGMBP 1-111#
PGMCD 1-113#
PGMFP 1-110#
PGMLEN 1-108#
PGMLNK 1-112#
PGMPTR 1-48#
PGMTG 1-128#
PIAEOI 1-286# 2-22% 2-26% 2-28% 2-101% 2-106% 2-111% 2-115% 2-131% 2-137% 2-139% 3-21
PIALT 1-292# 3-4# 3-14 7-35 7-37% 7-129 7-131%
PIASRQ 1-285# 2-16 2-34% 2-46% 2-65 2-67% 2-68 2-72% 2-79% 2-80 2-102 2-107 2-112 2-133%
2-135% 2-141% 2-153% 2-158% 2-165 2-174 2-176% 2-183 2-193 4-6 6-26%
PLOSTG 1-123#
PNDEOF 1-45#
PNDFLG 1-44# 3-17 3-19% 4-12 4-14%
PNTNTG 1-132#
PNTSTG 1-130#
POINT 1-241#
POLER 7-18 7-23 7-25 7-28#
POLERR 7-68 7-79 7-98#
POLERX 7-84 7-97 7-100#
POLEX 7-27 7-29#
POLEXT 7-65 7-92 7-103#
POLIT 7-74 7-76 7-82#
POLL 7-6# 7-35#
POLOOP 7-66# 7-95
POLVAL 7-11# 7-66 7-77
PPMODE 1-224#
PRIDEF 1-206#
PRTTG 1-141#
PSCTG 1-131#
PULFPN 1-298# 7-162
R0 1-38# 7-40% 7-45 7-46 7-58% 7-115% 7-118 7-122% 7-125 7-128% 7-155% 7-161
R1 1-38# 7-62% 7-82% 7-101% 7-110
R10 1-38#
R11 1-38# 7-41% 7-57
R12 1-38#
R13 1-38#
R14 1-38#
R15 1-38#
R16 1-38#
R17 1-38#
R18 1-39#
R19 1-39#
R2 1-38#
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#
R9 1-38#
RECLFG 1-217#
RFAIL 1-152#

CROSS REFERENCE TABLE (CREF V01-03)

```
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNTG  1-140#
RUNFLG  1-32#
RUNN    1-218#
SBP     1-47#
SCALER  1-82#   7-150
SCRMS   6-3#    6-43
SECDEF  1-205#
SENITG  1-144#  7-75    7-126
SNDIT   1-228#  5-28
SPOLON  7-67    7-177#  7-178#
SRQDID  4-8     4-11    4-15#
SRQRDY  2-47    4-2#    4-6#
STATSZ  1-247#
STKBLD  1-293#  7-59
STOVAL  7-111   7-114   7-140#
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSERR  1-43#
T51DEV  1-261#
TABPTR  1-243#
TAGPTR  1-239#
TCOL    1-245#  7-48*   7-50*   7-51    7-54    7-112
TRCFLG  1-34#
UNDEF   1-81#   7-152
VALTG   1-134#
VALUND  1-88#
X2      2-167   2-176#  2-187   2-195
XAXIS   1-248#  6-3#
```

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-38 7-147

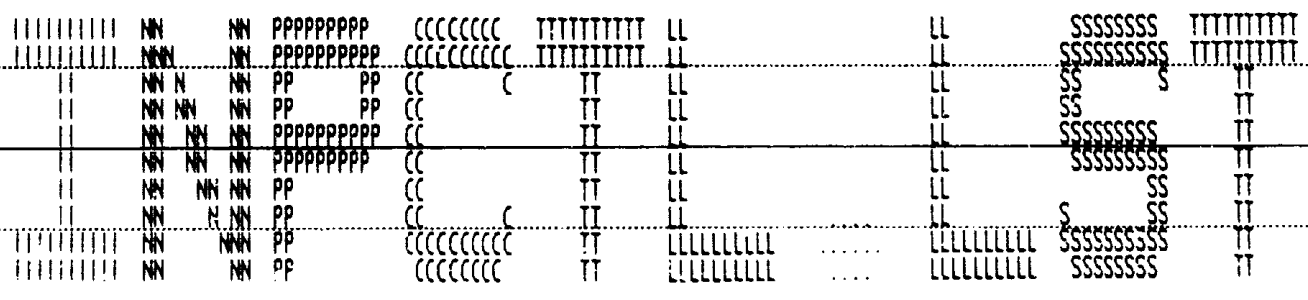


TABLE OF CONTENTS

```
                                .TITLE  INPCTL ROUTINES FOR I/O PROCESSOR
                                        IDENT /SBE024/
                                .GLOBL  INPCTL
                                .GLOBL  NEWBFR
0000'                           INPCTL=.
                                .GLOBL  INPSTG          ; INPUT TO STRING
                                .GLOBL  INPVAL  ; INPUT TO VALUE
                                .GLOBL  GETCHR          ; GETS A CHARACTER FORM THE BUFFER
                                .GLOBL  SCRTCH
                                .GLOBL  HCTRCS
                                .GLOBL  ASX
                                .GLOBL  KEYIN           ; KEYBOARD INPUT
                                .GLOBL  EFLAG           ; "E" FLAG FROM ASCII-->FPN
                        .SBTTL  INPUT--GET STRING
                        ;
                        ;
                        ; PURPOSE---1) FORM A COMPLETE ASCII STRING IN A STRING VARIABLE
                        ;           2) FORM A COMPLETE ASCII STRING FOR PROGRAM ANALYSIS.
                        ;
                        ;           INPUT OF DATA IS CONSIDERED COMPLETE WHEN EITHER OF THE
                        ;           FOLLOWING CONDITIONS ARE SATISFIED:
                        ;
                        ;           A) A LINE DELIMITER IS ENCOUNTERED
                        ;                RETURN
                        ;                DC3(  MODE)
                        ;                EOL
                        ;                EOF
                        ;                EOT
                        ;                ESC(  MODE)
                        ;           B) THE REQUESTED LENGTH OF THE STRING HAS BEEN INPUTTED.  IN
                        ;              THIS CASE THE I/O POINTER IS POSITIONED PAST THE NEXT
                        ;              DELIMITER, THIS INCLUDES ANY OF THE ABOVE PLUS A COMMA
                        ;
                        ; ENTER WITH-- POINT  =  POSITION DATA TO BE STORED
                        ;              LENGTH =  MAXIMUM NUMBER OF CHARACTERS TO BE READ
                        ;                        INTO THE STRING
                        ;                MODE =  FLAG INDICATING PAPER TAPE MODE
                        ;
                        ; EXIT WITH--- CHRCNT =  ACTUAL NUMBER OF CHARACTERS READ
                        ;              **** ERRCD POTENTIALLY SET BY I/O ERROR ****
                        ;
                        ; POINTERS, ETC.
                        ;.GLOBL LENGTH,PPMODE,CHRCNT,POINT
                        ;
                        ;SUBROUTINES
                        .GLOBL  GETCHR
0000    7F      0000G   INPSTG: CLR     COLCNT          ; EXIT FLAG BYTE
                                                        ; 0 - IF PROCESSING 1 - IF SKIPPING TO
                                                        ; DELIMITTER
0003    96      01G             LDA A   POINT+1,D       ; END < POINT + LENGTH (MAXIMUM POINT)
0005    9B      01G             ADD A   LENGTH+1,D
0007    97      01G             STA A   TCOL+1,D        ; END IS IN TCOL
0009    96      00G             LDA A   POINT,D
000B    99      00G             ADC A   LENGTH,D
000D    97      00G             STA A   TCOL,D
000F    BD      0000'   NXCHR:  JSR     GETCHR          ; GET A CHARACTER > A
0012    D6      00G             LDA B   CRSTAT,D        ; END CHARACTER FOUND?
0014    2B      14              BMI     EXIT1
```

INPUT--GET STRING

```
 331  0016  D6  00G          LDA B  COLCNT,D     ; EXIT SET?
 332  0018  26  F5           BNE    NXCHR        ; THEN SKIP CHARACTERS
 333  001A  DE  00G          LDX    POINT,D      ; qOINT CHAR
 334  001C  A7  00           STA A  0,X
 335  001E  08               INX                 ; POINT  POINT+1
 336  001F  DF  00G          STX    POINT,D      ; UPDATE POINTER INTO VARIABLE
 337  0021  9C  00G          CPX    TCOL,D       ; POINT = END
 338  0023  26  EA           BNE    NXCHR
 339  0025  7C  0000G        INC    COLCNT       ; SET EXIT
 340  0028  20  E5           BRA    NXCHR
 341  002A  96  0'G    EXIT1: LDA A  LENGTH+1,D   ; LENGTH - END
 342  002C  D6  00G          LDA B  LENGTH,D
 343  002E  90  01G          SUB A  TCOL+1,D
 344  0030  D2  00G          SBC B  TCOL,D
 345  0032  98  01G          ADD A  POINT+1,D    ; (LENGTH - END) + POINT
 346  0034  D9  00G          ADC B  POINT,D
 347  0036  97  01G          STA A  CHRCNT+1,D   ; RESULT ) CHRCNT
 348  0038  D7  00G          STA B  CHRCNT,D
 349  003A  39               RTS
```

```
                                .SBTTL  INPUT - GET VALUE
                                ;
                                ;
                                ; PURPOSE-----INPUT THE NEXT NUMERIC VALUE FROM THE PRESENT I/O DEVICE
                                ;             AND STORE THE RESULT IN THE 8 BYTE VARIABLE SPACE
                                ;             INDICATED.
                                ;
                                ; ENTER WITH--POINT  =  POSITION AT WHICH RESULTANT VALUE IS TO BE
                                ;                       STORED
                                ;            DEVICE  =  PRESENT I/O DEVICE (USED TO TEST FOR DISPLAY ONLY)
                                ;             MODE   =  FLAG INDICATING PAPER TAPE MODE INPUT
                                ;
                                ; EXIT WITH---ERRCD POTENTIALLY SET BY I/O ERROR OR CONVERSION ERROR.
                                ;
                                ; GLOBAL POINTERS
                                .GLOBL POINT,A.PRIM,PPMODE
                                ;SUBROUTINE CALLS
                                        .GLOBL  PULFPN,A8X
                                .GLOBL DSPGIN,ASCFPN,DIGFLG
                                ;
003B  96  000G        INPVAL:   LDA A   A.PRIM,D        ; DOING INPUT FROM DISPLAY (GIN)
003D  81  20                    CMP A   DSPDEV,I
003F  26  08                    BNE     DLIMCK
0041  7E  0127'                 JMP     DSPGIN          ; DO GIN FUNCTION
0044  30                TRYAGN: TSX                     ; CLEAN UP STACK OF BAD FP NUMBER
0045  BD  0000G                 JSR     A9X
0048  35                       TXS
0049  BD  008D'        DLIMCK: JSR     GETCHR          ; GET A CHARACTER
004C  D6  000G                  LDA B   CRSTAT,D        ; A DELIMITER?
004E  2A  05                    BPL     LEGLCK
0050  C5  38                    BIT B   EJETYP,I        ; CHECK FOR EOF (ALSO ERRORS)
0052  27  F5                    BEQ     DLIMCK          ; GO FOR MORE CHARACTERS IF OK
0054  39                       RTS                     ; ELSE RETURN
0055  CE  007F'        LEGLCK: LDX     LEGTAB,I
0058  E6  00           1$:     LDA B   0,X             ; GET LEGAL CHAR FROM TABLE
005A  27  ED                    BEQ     DLIMCK          ; 0 MARKS END OF TABLE
005C  11                        CBA
005D  27  03                    BEQ     CHAROK
005F  08                        INX
0060  20  F6                    BRA     1$
0062  BD  0000G         CHAROK: JSR     ASCFPN          ; ASCII -> FPN
0065  DE  000G                  LDX     A.PTR,D         ; NEED TO BACK UP PAST DELIM
0067  09                       DEX
0068  B6  0000G                 LDA A   EFLAG           ; IF NO NUMBER AFTER "E" THEN NEED MORE BACK
006B  27  05                    BEQ     1$
006D  09                       DEX
006E  86  45                    LDA A   'E,I            ; IN FACT NEED TO PUT IT BACK
0070  A7  00                    STA A   0,X
0072  DF  000G          1$:     STX     A.PTR,D
0074  70  0000G                 TST     DIGFLG          ; SEE IF ANY DIGITS DETECTED
0077  26  CB                    BNE     TRYAGN
0079  DE  000G                  LDX     POINT,D
007B  BD  0000G                 JSR     PULFPN          ; TRANSFER RESULT TO POINT
007E  39                       RTS
007F  30  31    32  LEGTAB:    .ASCII/0123456789/      ; LEGAL CHAR TABLE
0082  33  34  35
```

INPLIT - GET VALUE

```
                0085    36      37      38
                0088    39
    57          0089    2B      2D      2E          .ASCIZ/+-./
                008C    00
```

```
                                .SBTTL  GET CHARACTER
                                ; PURPOSE---- RETRIEVES AN INPUT BYTE FROM WHAT EVER DEVICE IS ACTIVE
                                ;
                                ; ENTER WITH- A.PTR  =  POSITION OF NEXT BYTE IN BUFFER (CALLS FOR
                                ;                       NEW BUFFER IF NECESSARY.)
                                ;              A.END  = POSITION OF LAST VALID CHARACTER+1 IN THE
                                ;                       PRESENT BUFFER (END OF LINE IS NOT A VALID CHAR.)
                                ;
                                ;
                                ; EXIT WITH-- CHAR   =  LAST CHARACTER
                                ;            ACC A  =  LAST CHARACTER
                                ;            A.PTR  =  SET FOR NEXT CHARACTER POSITION
                                ;       **** ERRCD BYTE POTENTIALLY SET BY I/O ROUTINES ****
                                ;
                                ;
                                ;---------- THE INPUT BUFFER ROUTINE WAS COMBINED WITH GET CHAR------
                                ;
                                ;PURPOSE---- INPUT AN ASCII BUFFER AND REPOSITION BUFFER POINTERS
                                ;
                                ; ENTER WITH- IOFUNC  =  TYPE OF I/O OPERATION
                                ;            PPMODE  =    MODE(1)  OR  MODE(0)  FOR INPUT
                                ;            A.PRIM  =  I/O DEVICE
                                ;              A.STRT  = START OF PERTINENT BUFFER
                                ;              A.MAX   = LAST AVAILABLE POSITION IN BUFFER
                                ;
                                ;EXIT WITH-- A.PTR   =  POINTER TO FIRST CHARACTER IN BUFFER
                                ;            A.END   =  LAST VALID CHARACTER+1
                                ;       ***** ERRCD BYTE POTENTIALLY SET BY I/O ROUTINES ****
                                ;
                                ;
                                ;*****************************************************************************
                                ;       SUBROUTINES TO INPUT BUFFERS
                                .GLOBL  FILEIN,KEYIN,IECIN,XFRCTL
                                        .GLOBL  PULFPN,PSHFPN
                                ;
008D  D6  000   GETCHR: LDA B   A.STAT,D        ; GET STATUS
008F  C5  20            BIT B   BFRSTT,I        ; TEST BUFFER STATUS
0091  27  0E            BEQ     NWBFR           ; GO GET A NEW BUFFER
0093  DE  000   BFROK:  LDX     A.PTR,D         ; GET POINTER
0095  9C  000           CPX     A.END,D         ; TEST IF TRYING TO REACH PAST END
0097  27  0E            BEQ     ENDBFR
0099  A6  00            LDA A   0,X             ; GET CHAR. FROM BUFFER
0098  97  000   BFR.1:  STA A   CHAR,D          ; SAVE IT
009D  08                INX                     ; SET POINTER TO NEXT POSITION
009E  DF  000           STX     A.PTR,D
00A0  39                RTS
00A1  BD  00B9' NWBFR:  JSR     NEWBFR          ; GO GET BUFFER
00A4  4D                TST A                   ; ERRORS?
00A5  27  EC            BEQ     BFROK
                        ;*****************************************************************************
                        ;       DO BUFFER EMPTY SPECIAL STUFF
00A7  96  000   ENDBFR: LDA A   CRSTAT,D        ; GET CR STATUS
00A9  27  F6            BEQ     NWBFR           ; IF = 0 THEN OK TO GET A NEW BUFFER
00AB  8A  80            ORA A   CRVLD,I         ; MAKE CR STATUS VALID
00AD  97  000           STA A   CRSTAT,D
00AF  D6  000           LDA B   A.STAT,D
00B1  C4  DF            AND B   -1-BFRSTT,I     ; CLEAR VALID BUFFER FLAG
```

```
                                        STA B   A.STAT,D
                                        LDA A   15.,I           ; GET "CR" FOR EVERYONE TO DELIMIT ON!
                                        BRA     BFR.1           ; EXIT
                                ;*******************************************************************
                                ;       GET A NEW BUFFER
                                NEWBFR:
                                        JSR     NXBFR           ; GO GET NEW BUFFER
                                        LDA B   PPMODE.D        ;IF NOT  MODE THEN DON'T SLOW DOWN
                                        BEQ     NEWBEX          ; SO RETURN
                                        LDX     A.PTR,D         ; ELSE SCAN BUFFER FOR A "XCR"
                                        LDA B   15.,I
                                1$:     CMP B   0,X
                                        BEQ     PURGE           ; IF FIND "CR" IN PRESENT BUFFER
                                        CPX     A.END,D         ;       PURGE TILL FIND EOL CHAR
                                        BEQ     NEWBEX
                                        INX
                                        BRA     1$
                                PURGE:  STX     A.END,D         ; SET POSITION OF "CR"
                                PURG.:  LDA B   CRSTAT,D
                                        BNE     NEWBEX          ; EXIT IF EOL OR EQUIVALENT IN BUFFER
                                        LDX     A.STAT,!        ; ELSE DO DUMMY READS
                                        JSR     PSHFPN          ; SAVE STATUS
                                        LDX     SCRTCH.I        ; READ INTO SCRATCH
                                        STX     A.STRT,D
                                        LDX     #CTRCS-3,I      ; WHICH HAS GOT TO BE BIG
                                        STX     A.MAX,D         ; ENOUGH TO HOLD JUST ABOUT EVERYTHING
                                        JSR     NXBFR           ; GET THIS BUFFER
                                        LDX     A.STAT,I        ; RESTORE STATUS
                                        JSR     PULFPN
                                        LDX     A.STRT,D
                                        STX     A.PTR,D
                                        LDA A   ERRCD,D
                                        BRA     PURG.           ; TRY AGAIN
                                NEWBEX: RTS                     ;A RETURN FOR GOOD MEASURE
                                ;
                                ;       REALLY GETS THE BUFFER
                                ;
                                NXBFR:  CLR     CRSTAT          ; INITIALIZE THE WORLD
                                        LDX     A.STRT,D        ; SET POINTER TO BEGINNING OF BUFFER
                                        STX     A.PTR,D
                                        LDX     IOTBL,I         ; GET I/O DISPATCH TABLE
                                        JSR     XFRCTL          ; GO PREFORM THIS MIRICAL
                                        LDX     A.STRT,D        ; GET START OF BUFFER
                                        STX     A.PTR,D         ; SET UP POINTER
                                        LDA A   ERRCD,D         ; TEST FOR ERROR
                                        BEQ     1$
                                        LDA B   CREOF+CRVLD,I   ; FORCE END
                                        STA B   CRSTAT,D
                                1$:     LDA B   A.STAT,D                ; MARK BUFFER VALID
                                        ORA B   BFRSTI,I
                                        STA B   A.STAT,D
                                                ; GO SET UP CHAR.
                                        RTS
                                ;*******************************************************************
                                ;       INPUT BUFFER ROUTINE TABLE (BY DEVICE TYPE)
                                        .GLOBL  MTPIN
                                IOTBL:  .WORD   FILEIN          ; FILE INPUT
```

```
115     0119    0000G                   .WORD   IECIN           ; IEC PORT INPUT
116     0118    0168'                   .WORD   KEYIN           ; KEYBOARD INPUT
117     0110    0000G                   .WORD   NIODEV          ; DISPLAY INPUT
118     011F    0000G                   .WORD   NIODEV          ; MAG TAPE INPUT
119     0121    0000G                   .WORD   NIODEV          ; INTERNAL XFER
120     0123    0000G                   .WORD   MTPIN           ; MAGTAPE INPUT DEV. 2
121     0125    0000G                   .WORD   NIODEV          ; INTERNAL TRANSFER OVER RUN
```

```
                                        .SBTTL  DSPGIN--DISPLAY INPUT FUNCTIONS
                                ;THERE ARE TWO DISPLAY INPUT FUNCTIONS. THEY ARE:
                                ;1) GIN---INPUT POSITION OF CURSOR POINT IN GDU'S
                                ;2) INPUT-INPUT SCREEN SIZE IN GDU'S  (USABLE FOR INFO ON ASPECT)
                                ;
                                        .GLOBL  DOFP,FHEX,FRET,FLOAT1,YAXIS
                0080                    FA = 200
                00A0                    FS = 240
                00C0                    FM = 300
                00E0                    FD = 340
                                        .GLOBL  TMPHY.TMPLY.TMPHX.TMPLX.GSCNUM
                                ;
0127    96      00G     DSPGIN: LDA A   A.SEC,D         ; SEE IF VALID I/O ADDRESS
0129    81      18              CMP A   24.,I           ; GIN
012B    27      1A              BEQ     GINIT
012D    81      0D              CMP A   13.,I           ; SIZE INPUT
012F    27      03              BEQ     DINPIT
0131    7E      0000G           JMP     NIODEV          ; ILLEGAL INPUT FUNCTION
0134    96      00G     DINPIT: LDA A   YAXIS,D         ; FIGURE OUT WHICH AXIS REQUESTED
0136    26      04              BNE     INPY
0138    86      82              LDA A   130.,I          ; X AXIS SIZE
013A    20      02              BRA     INPAXI
013C    86      64      INPY:   LDA A   100.,I          ; Y AXIS SIZE
013E    36              INPAXI: PSH A
013F    4F                      CLR A
0140    36                      PSH A
0141    36                      PSH A
0142    BD      0000G           JSR     FLOAT1          ; FLOAT IT
0145    20      1B              BRA     GINNXT          ; SAVE IT AND SET UP FOR NEXT
0147    96      00G     GINIT:  LDA A   YAXIS,D
0149    26      06              BNE     GINY
014B    96      00G             LDA A   TMPHX,D         ; GET XAXIS INFORMATION
014D    D6      00G             LDA B   TMPLX,D
014F    20      04              BRA     GINAXI
0151    96      00G     GINY:   LDA A   TMPHY,D
0153    D6      00G             LDA B   TMPLY,D
0155    37      GINAXI: PSH B                           ; FLOAT IT
0156    36                      PSH A
0157    36                      PSH A
0158    BD      0000G           JSR     FLOAT1
015B    BD      0000G           JSR     DOFP            ; CONVERT TO GDU'S
015E    E0G                     .BYTE   FHEX+FD
015F    0000G                   .WORD   GSCNUM
0161    00G                     .BYTE   FRET
0162    DE      00G     GINNXT: LDX     POINT,D         ; SAVE RESULT
0164    BD      0000G           JSR     PULFPN
0167    73      0000G           COM     YAXIS           ; SET UP FOR NEXT
016A    39                      RTS
                                ;
```

```
                                        .SBTTL  KEYIN INPUT OF A KEYBOARD BUFFER
                                ;THIS ROUTINE DOES THE IDLE LOOP WHILE WAITING FOR KEYBOARD
                                ; INPUT.   IT SETS APPROPRIATE STATUS FLAGS FOR THE KBPROC.
                                        .GLOBL  CIDLE           ; CURSOR IDLE LOOP
                                        .GLOBL  PNDFLG
                                        .GLOBL  ENDKEY
                                ;
                                        .GLOBL  N1ODEV
0168  96  00G     KEYIN:  LDA A   A.SEC,D          ; MAKE SURE LEGAL SEC. ADDRES
016D  81  0D            CMP A   13.,I
016F  27  04            BEQ     1$
0171  BD  0000G         JSR     N1ODEV          ; NO I/O DEVICE
0174  39               RTS
0175  86  3F      1$:   LDA A   '?,I            ;SET "?" AS THE CURSOR
0177  97  00G           STA A   CURSOR,D
0179  86  03            LDA A   INPUTK!RUNN,I   ; SET INPUT AND RUN MODE FOR KEYBOARD
017B  97  00G           STA A   KBFLAG,D
017D  BD  0000G         JSR     CIDLE
0180  D6  00G           LDA B   KBFLAG,D
0182  C5  40            BIT B   ENDKEY,I
0184  27  EF            BEQ     1$
0186  C6  02            LDA B   RUNN,I          ; RESET KEYBOARD
0188  D7  00G           STA B   KBFLAG,D
018A  86  80            LDA A    H80,I  ; RESET BLOT AS CURSOR
018C  97  00G           STA A   CURSOR,D
018E  39                RTS
              0001'             .END
```

SYMBOL TABLE

```
ABRFLG=  0040        AFAIL =  0030        A1AT  =  0010        ARRAY =  0020        ASCFPN= XXXXXX G
ASTR  =  0020        ATLOD =  0008        ATSNTG= XXXXXX G     ATVLD =  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     A8X   = XXXXXX G     A9X   = XXXXXX G     BAKSTG= XXXXXX G     BANK  = XXXXXX G
BFROK    0093R       BFRSTI=  0020        BFR.1    0098R       BLINK = XXXXXX G     BMAT  =  0004
BRKCNT= XXXXXX G     BSTR  =  0008        BUSACT=  0010        CDOPTR= XXXXXX G     CDSPTR= XXXXXX G
CHAR  = XXXXXX G     CHAROK   0062R       CHRCNT= XXXXXX G     CIDLE = XXXXXX G     CLPTR = XXXXXX G
CMAT  =  0001        COLCNT= XXXXXX G     CRDC3 =  0002        CREOF =  0008        CREOI =  0004
CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT= XXXXXX G     CRVLD =  0080
CSTR  =  0002        CTKN  = XXXXXX G     CURSOR= XXXXXX G     DATDEV=  0022        DIGFLG= XXXXXX G
DIMFLG=  0004        DINPIT   0134R       DIRCT =  0080        DISCNT= XXXXXX G     DISSRQ=  0008
DL    = XXXXXX G     DLIMCK   0049R       DOFP  = XXXXXX G     DP    = XXXXXX G     DREXTA= XXXXXX G
DREXTB= XXXXXX G     DSPDEV=  0020        DSPGIN   0127RG      DT    = XXXXXX G     EDTBFR= XXXXXX G
EFLAG = XXXXXX G     ENDBFR   0067R       ENDKEY=  0040 G      EOETYP=  0038        EOLTG = XXXXXX G
EOSTG = XXXXXX G     ERATSN= XXXXXX G     ERDOMN= XXXXXX G     EREOM = 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     EXIT1    002AR       EXTFLG=  0080
F2    =  0080        FD    =  00E0        FHEX  = XXXXXX G     FILDEV=  0000        FILEIN= XXXXXX G
FIX1A =  0003        FIX1B =  0004        FLOAT1= XXXXXX G     FM    =  00C0        FMTVLD=  0008
FNFLG =  0010        FORTG = XXXXXX G     FRET  = XXXXXX G     FS    =  0040        GETCHR   008DRG
GINAX1   0155R       GINIT    0147R       GINNXT   0162R       GINY     0151R       GLOFLG= XXXXXX G
GOSTG = XXXXXX G     GSCNUM= XXXXXX G     HCTRCS= XXXXXX G     IECIN = XXXXXX G     IMXTG = XXXXXX G
INPAX1   013ER       INPCTL=  0000RG      INPSTG   0000RG      INPUTK=  0001        INPVAL   003BRG
INPY     013CR       IOBFR1= XXXXXX G     IOFLGS= XXXXXX G     IOFUNC= XXXXXX G     IOTBL    0117R
ITM1TG= XXXXXX G     ITM2TG= XXXXXX G     ITTDEV=  0024        JMPX  = XXXXXX G     KBDEV =  001F
KBFLAG= XXXXXX G     KBIN  = XXXXXX G     KEYFLG=  0010        KEYIN    016ARG      KEYSTK= XXXXXX G
LBRKTG= XXXXXX G     LCLFLG= XXXXXX G     LDAX  = XXXXXX G     LDXX  = XXXXXX G     LEGLCK   0055R
LEGTAB   007FR       LENGTH= XXXXXX G     LISTTG= XXXXXX G     LNNOTG= XXXXXX G     LOCTG = XXXXXX G
LSP   = XXXXXX G     LSTEMT=  0002        MTBFR = XXXXXX G     MTPDEV=  0021        MTPD2 =  0023
MTPIN = XXXXXX G     NEWBEX   00F6R       NEWBFR   0089RG      NIODEV= XXXXXX G     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=  0005        NULLTG= XXXXXX G     NUMBFR   0091R       NXBFR    00F7R       NXCHR    000FR
OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002
OPRADR= XXXXXX G     PAETG = XXXXXX G     PARM  =  0008        PGMATR=  0002        PGMBP =  0005
PGMCD =  0009        PGMEP =  0003        PGMLEN=  0000        PGMLNN=  0007        PGMPTR= XXXXXX G
PGMTG = XXXXXX G     PLOSTG= XXXXXX G     PNDEOF= XXXXXX G     PNDFLG= XXXXXX G     PNTNTG= XXXXXX G
PNTSTG= XXXXXX G     POINT = XXXXXX G     PPMODE= XXXXXX G     PRIDEF=  0001        PRTTG = XXXXXX G
PSCTG = XXXXXX G     PSHFPN= XXXXXX G     PULFPN= XXXXXX G     PURGE    00CER       PURG     00D1R
RECLFG=  0004        RFAIL =  00C0        RMAT  =  0040        RPTCTL=  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        SCRTCH= XXXXXX G
SECDEF=  0002        SEMITG= XXXXXX G     SNDIT =  0080        STAT37= XXXXXX G     STPFLG=  0040
STPKEY=  0020        STRING=  0010        SYSERR= XXXXXX G     TABPTR= XXXXXX G     TAGPTR= XXXXXX G
TCOL  = XXXXXX G     TMPHX = XXXXXX G     TMPHY = XXXXXX G     TMPLX = XXXXXX G     TMPLY = XXXXXX G
TRCFLG=  0020        TRYAGN   0044R       T51DEV=  0025        UNDEF =  0080        VALTG = XXXXXX G
VALUND=  0040        XAXIS = XXXXXX G     XFRCTL= XXXXXX G     YAXIS = XXXXXX G
. ABS.   0000    00
         018F    01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2402  WORDS
,SY:INPCTL/C<DK1:SETCLI,INPCTL
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

```
INPSTG  1-279#  1-319#
INPUTK  1-219#  5-16
INPVAL  1-280#  2-22#
INPY    4-20    4-23#
IOBFR1  1-266#
IOFLGS  1-246#
IOFUNC  1-190#
IOTBL   3-98    3-114#
ITM1TG  1-136#
ITM2TG  1-132#
ITTDEV  1-260#
JMP%    1-55#
KBDEV   1-255#
KBFLAG  1-210#  5-17*  5-19  5-23*
KBIN    1-208#
KEYFLG  1-35#
KEYIN   1-285#  3-33#  3-116  5-9#
KEYSTK  1-52#
LBRKTG  1-139#
LCLFLG  1-24#
LDAX    1-65#
LDXX    1-60#
LEGLCK  2-31    2-35#
LEGTAB  2-35    2-56#
LENGTH  1-240#  1-323  1-326  1-341  1-342
LISTTG  1-127#
LNNOTG  1-135#
LOCTG   1-145#
LSP     1-46#
LSTFMT  1-227#
MTBFR   1-262#
MTPD2   1-259#
MTPDEV  1-257#
MTPIN   3-113#  3-120
NEWBEX  3-66    3-72  3-77  3-91#
NEWBFR  1-277#  3-47  3-63#
NIODEV  3-117   3-118  3-119  3-121  5-18  5-8#  5-12
NLPTR   1-21#
NOKEY   1-209#
NOOUT   1-225#
NOWRTY  1-226#
NTAPTR  1-93#
NTATTR  1-80#
NTDTMS  1-92#
NTDLEN  1-95#
NTLINK  1-77#
NTNAME  1-78#
NTPTR   1-22#
NTRELY  1-215#
NTSPTR  1-97#
NTVAL   1-87#
NTWCOL  1-91#
NTWLEN  1-96#
NTWROW  1-90#
NULLTG  1-122#
NWBFR   3-38    3-47#  3-53
NXBFR   3-64    3-84   3-95#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
NXCHR   1-328#  1-332   1-338   1-340
OBJATR  1-102#
OBJBCK  1-103#
OBJDT   1-104#
OBJLEN  1-101#
ONSFLG  1-29#
OPRADR  1-238#
PRETG   1-133#
PARM    1-85#
PGMATR  1-109#
PGMBP   1-111#
PGMCD   1-113#
PGMEP   1-110#
PGMLEN  1-108#
PGMLNN  1-112#
PGMPTR  1-48#
PGMTG   1-128#
PLOSTG  1-123#
PNDEOF  1-45#
PNDFLG  1-44#   5-5#
PNTNTG  1-132#
PNTSTG  1-130#
POINT   1-241#  1-322   1-325   1-333   1-336#  1-345   1-346   2-53    4-45
PPMODE  1-224#  3-65
PRIDEF  1-206#
PRTTG   1-141#
PSCTG   1-131#
PSHFPN  3-34#   3-79
PULFPN  2-19#   2-54    3-34#   3-86    4-46
PURG    3-76#   3-90
PURGE   3-70    3-75#
R0      1-38#
R1      1-38#
R10     1-38#
R11     1-38#
R12     1-38#
R13     1-38#
R14     1-38#
R15     1-38#
R16     1-38#
R17     1-39#
R18     1-39#
R19     1-39#
R2      1-38#
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#
R9      1-38#
RECLFG  1-217#
RFAIL   1-152#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNIG  1-140#
RUNFLG  1-32#
RUNN    1-218#  5-16    5-22
SBP     1-47#
SCALER  1-82#
SCRTCH  1-282#  3-80
SECDEF  1-206#
SEMITG  1-144#
SNDIT   1-228#
STATST  1-247#
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSERR  1-43#
T51DEV  1-261#
TABPTR  1-243#
TAGPTR  1-239#
TCOL    1-245#  1-324*  1-327*  1-337   1-343   1-344
TMPHX   4-11#   4-32
TMPHY   4-11#   4-35
TMPLX   4-11#   4-33
TMPLY   4-11#   4-36
TRCFLG  1-34#
TRYAGN  2-26#   2-52
UNDEF   1-81#
VALTG   1-134#
VALUND  1-88#
XAXIS   1-248#
XFRCTL  3-33#   3-99
YAXIS   4-6#    4-19    4-30    4-47*
```

CROSS REFERENCE TABLE (CREF V01-03.)

SEI 1-7#

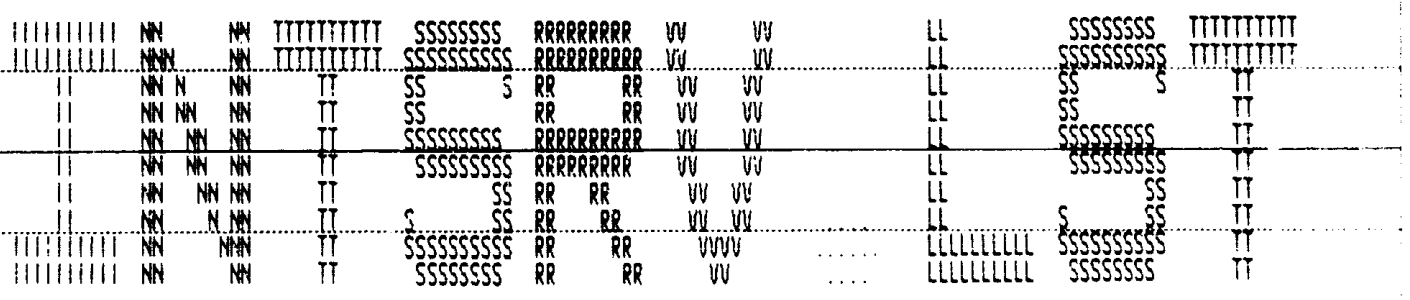


TABLE OF CONTENTS

```
                                .TITLE  INTSRV  INTERUPT SERVICE ROUTINE
0000'                           .CSECT  INTSRV
                                .IDENT  /SB2004/
                                .GLOBL  PIATBL          ; PIA TABLE
                                .GLOBL  INIQQQ          ; PLAYGROUND
                                                        ; FORM: 2 BYTES PIA ADDRESS
                                                        ;       1 BYTE BANK (MSB SET IF ACIA)
                                                        ;       2 BYTES SERVICE ROUTINE ADDRESS
                                .GLOBL  BANKSW          ; HARDWARE BANK SWITCH REG.
                                .GLOBL  BANK            ; SOFTWARE BANK REG.
                                .GLOBL  INTSRV          ; INTERUPT VECTOR (HARDWARE) POINTS HERE
```

```
                                        .SBTTL  INTSRV--INTERUPT SERVICE ROUTINE
                                ;THIS ROUTINE SERCHES THE PIA'S FOR HOREWARE INTERUPTS AND
                                ;CALLS THE APROPRIATE SERVICE ROUTINE WHICH MUST RETURN HERE TO
                                ;PUT THINGS BACK TOGETHER AGAIN.
                                ;  THIS IS THE FIXED VERSION THAT ALLOWS THE USE
                                ; OF THE BANK EXPANDER BOX.
                                ;
                                        .GLOBL  SETBNK
                                ;
0000   96   00G      INTSRV: LDA A   BANK,D          ; SAVE PRESENT BANK
0002   36                   PSH A
0003   CE   0000G            LDX     PIATBL,I        ; GET START OF TABLE
0006   DF   00G      POLLOP: STX     INTQQQ,D        ; SAVE INDEX FOR AWHILE
0008   A6   02               LDA A   2,X             ; GET BANK
000A   27   05               BEQ     1$              ; IF SYSTEM PIA DON'T SWITCH
000C   84   3F               AND A   #3F,I           ; CLEAR OFF GARBAGE BITS
000E   BD   0000G            JSR     SETBNK          ; SWITCH BANKS
0011   EE   00       1$:     LDX     0,X             ; GET PIA ADDRESS
0013   27   21               BEQ     INTEXT          ; IF ZERO THEN DONE
0015   E6   01               LDA B   1,X             ; GET INTERRUPT REG. OF PIA
0017   DE   00G              LDX     INTQQQ,D        ; GET BANK BACK AGAIN
0019   A6   02               LDA A   2,X
001B   2B   12               BMI     ACIA            ; WAS IT AN ACIA REFERENCE?
001D   C8   0F               EOR B   #00F,I          ; ELSE TEST FOR PIA INTERRUPT
001F   C5   81               BIT B   #81,I
0021   27   0F               BEQ     DSPAT
0023   C5   68               BIT B   #68,I
0025   27   08               BEQ     DSPAT
0027   0004         NOTINT: .REPT   6
                            .LIST   MEB
                            INX
                            .ENDM
                            INX
                            INX
                            INX
                            INX
                            INX
                            INX
002D   20   D7               BRA     POLLOP
002F   5D           ACIA:   TST B
0030   2A   F5               BPL     NOTINT          ; NOPE
0032   EE   03       DSPAT:  LDX     3,X             ; JUMP TO SERVICE ROUTINE
0034   AD   00               JSR     0,X
0036   32           INTEXT: PUL A                   ; GET OLD BANK FROM STACK
0037   84   3F               AND A   #03F,I          ; STRIP OFF OLD RATE
0039   BD   0000G            JSR     SETBNK          ; RESET BANK
003C   3B                   RTI                     ; RETURN FROM INTERRUPT
                            .BLKB   2               ; SPACE FILLER
0001'                       .END
```

```
INTSRV INTERUPT SERVICE ROUTIN RT-11 MMAC  VM02-10  14-OCT-76 01:34:47 PAGE 2+
SYMBOL TABLE

AC1A     002FR     02 BANK  = XXXXXX G       BANKSW= XXXXXX G       DSPAT    0032R     02 INTEXT   0036R     02
INTQQQ= XXXXXX G       INTSRV   0000RG    02 NOTINT   0027R     02 PIATBL= XXXXXX G       POLLOP   0006R     02
SETBNK= XXXXXX G
. ABS.   0000      00
         0000      01
INTSRV   003F      02
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3419.  WORDS
.SY:INTSRV/C/DK1:SE1CL1.INTSRV
```

CROSS REFERENCE TABLE (CREF V01-03)

ACIA	2-23	2-34#	
BANK	1-25#	2-10	
BANKSW	1-24#		
DSPAT	2-26	2-28	2-36#
INTEXT	2-19	2-38#	
INTQQQ	1-20#	2-13*	2-21
INTSRV	1-26#	2-10#	
NOTINT	2-29#	2-35	
PIATBL	1-19#	2-12	
POLLOP	2-13#	2-33	
SETBNK	2-8#	2-17	2-40

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3

```
 IIIIIIIIII   00000000   CCCCCCCC  TTTTTTTTTT  LL                          LL          SSSSSSSS  TTTTTTTTTT
 IIIIIIIIII  0000000000  CCCCCCCCCC TTTTTTTTTT  LL                          LL         SSSSSSSSSS TTTTTTTTTT
    II      00      00   CC      C     TT      LL                          LL         SS      S      TT
    II      00      00   CC            TT      LL                          LL         SS             TT
    II      00      00   CC            TT      LL                          LL         SSSSSSSS       TT
    II      00      00   CC            TT      LL                          LL          SSSSSSSS      TT
    II      00      00   CC            TT      LL                          LL                SS      TT
    II      00      00   CC      C     TT      LL                          LL         S      SS      TT
 IIIIIIIIII  0000000000  CCCCCCCCCC    TT      LLLLLLLLLL  ......          LLLLLLLLLL SSSSSSSSSS     TT
 IIIIIIIIII   00000000   CCCCCCCC      TT      LLLLLLLLLL  ....            LLLLLLLLLL  SSSSSSSS      TT
```

TABLE OF CONTENTS

```
                                .TITLE  IOCTL   I/O CONTROL ROUTINES
                                .IDENT  /SBL062/
0000'                           .CSECT  IOCTL
                                .GLOBL  SCRTCH          ; DAKINS' PATCH
                                .GLOBL  ADRDEV          ; ADDRESS A DEVICE
                                .GLOBL  BFRALC          ; ALLOCATE IT A BUFFER
                                .GLOBL  UNADR           ; UNADDRESS A DEVICE
                                .GLOBL  OPRADR          ; OPERATION CODE REG. FROM EVAL
                                .GLOBL  ATPROC          ;  PROCESSOR (  ALSO)
                                .GLOBL  ATNON           ; ATTENTION ON (IEC)
                                .GLOBL  IECSND          ; SEND BYTE ON IEC BUS
                                .GLOBL  INTSRC          ; INITIALIZE IEC AS SOURCE
                                .GLOBL  STKBLD          ; REVERSE STACK BUILDER
                                .GLOBL  DSPSTT          ; DISPLAY STATUS
                                .GLOBL  PIALX,PIAHX,PIALY,PIAHY ; DISPLAY PIA'S
                                .GLOBL  PUPTAP          ; POWER UP MAGTAPE
                                .GLOBL  MAGEND          ; MAGTAPE DONE FLAG
                                .GLOBL  CRTRST          ; RESET CRT DRIVERS
                                .GLOBL  FPC             ; TEMP FP. C
                                .GLOBL  PULFPN
                                .GLOBL  ISL,ISP         ; IMAGE POINTERS
                                .GLOBL  ERNFST          ; NO FORMAT STATEMENT ERROR
```

```
                                .SBTTL  ADRDEV  ADDRESS DEVICE CONTROLLER
                                ; THIS ROUTINE ASSIGNS DEFAULT PRIMARY AND
                                ; SECONDARY ADDRESSES FOR CHANNEL A IF THEY HAVE
                                ; NOT BEEN SET BY  OR # PREVIOUSLY.
                                ;
                                        .GLOBL  PIALT, POINTB, DB
                                        .GLOBL  IOSYSF
                                        .GLOBL  IECRLF          ; CR OR CR/LF FORMAT FLAG
                                        .GLOBL  SCMSEO
                                        .GLOBL  PPNUL,NULCHR
                                        .GLOBL  PPEOR,PPEOF,EOLCHR,EIXCHR
                                        .GLOBL  TABCNT
                                        .GLOBL  MTPADR
                        0040            IOPLT   = 040           ; I/O IN PROGRESS LIGHT
                                ;
0000    86      000     ADRDEV: LDA A   RTRNTG,I        ; TAG IT
0002    36                      PSH A
0003    96      000             LDA A   OPRADR,D        ; GET DEFAULT SECONDARY & I/O FUNC FROM TABLE
0005    97      000             STA A   IOFUNC,D        ; SAVE ENTIRE CODE
0007    D6      000             LDA B   A.STAT,D        ; GET I/O STATUS FOR CHAN. A
0009    C4      CF              AND B   #-1-(BFRSTT!BUSACT),I   ; CLEAR SOME STATUS
000B    C5      04              BIT B   ATVLD,I         ;  INFO ON STACK?
000D    27      0E              BEQ     ADROK           ; ALL OK SO ADDRESS DEVICE
000F    F7      0000G           STA B   A.STAT          ; SAVE STATUS THUS FAR
0012    BD      0172'           JSR     ATPROC          ; GO PROCESS  STUFF ON STACK
0015    96      000             LDA A   ERRCD,D         ; ANY ERRORS?
0017    27      02              BEQ     1$
0019    32                      PUL A
001A    39                      RTS
001B    D6      000     1$:     LDA B   A.STAT,D        ; GET STATUS BACK AGAIN
001D    C5      02      ADROK:  BIT B   SECDEF,I        ; TEST IF ALREADY DEFINED
001F    26      06              BNE     PRI             ; IF SO TRY PRIMARY
0021    96      000             LDA A   IOFUNC,D        ; GET DEFAULT SEC. ADDRESS
0023    84      3F              AND A   #3F,I   ; MASK OUT SOME STATUS BITS
0025    97      000             STA A   A.SEC,D         ; STORE IN ADDRESS LOCATION
0027    C5      01      PRI:    BIT B   PRIDEF,I        ; TEST IF PRIMARY DEFINED
0029    26      06              BNE     DNADDR          ; THEN DONE WITH ADDRESSES
002B    96      01G             LDA A   OPRADR+1,D      ; GET PRIMARY INFO FROM TABLE
002D    84      3F              AND A   #3F,I   ; EXTRACT ADDRESS
002F    97      000             STA A   A.PRIM,D        ; SAVE IT IN PRIMARY ADDRESS.
0031    50              DNADDR: TST B                   ; IS DIRECTION INFO ALREADY SET UP?
0032    2B      04              BMI     1$
0034    96      01G             LDA A   OPRADR+1,D      ; GET CODE BACK AGAIN
0036    84      C0              AND A   #GC0,I  ; EXTRACT DIRECTION STATUS
0038    18              1$:     ABA                     ; OR IN DIRECTION WITH ISTATA
0039    8A      03              ORA A   #PRIDEF!SECDEF,I ; MAKE ALL DEFINED
003B    97      000             STA A   A.STAT,D        ; SAVE NEW STATUS
003D    96      000             LDA A   A.PRIM,D        ; PRIMARY ADDRESS
003F    81      21              CMP A   MTPDEV,I        ; IS IT MAG TAPE
0041    26      05              BNE     ADRMOR
0043    BD      0000G           JSR     MTPADR          ; MAGTAPE SPECIAL ADDRESSING STUFF
0046    20      08              BRA     ADRDON
0048    81      20      ADRMOR: CMP A   DSPDEV,I        ; DISPLAY NEEDS SOME HAND FEEDING
004A    27      0C              BEQ     ADRD
004C    81      22              CMP A   DATDEV,I        ; IF DATA I/O THEN NO LIGHT!!!!!!!!
004E    27      08              BEQ     ADRD
0050    86      0000G   ADRDON: LDA A   PIALT           ; TURN ON I/O LIGHT
```

ADRDEV ADDRESS DEVICE CONTROLLER

```
   58  0053  8A  40            ORA A  IOPLT,I
   59  0055  B7  0000G         STA A  PLALT
   60  0058  86  01      ADRD.: LDA A  1,I              ; INITIALIZE TAB COUNTER FOR J.K.
   61  005A  B7  0000G         STA A  TABCNT
   62  005D  B7  0000G         STA A  IOSYSF          ; I/O SYSTEM FLAG FOR FIRST ACCESS
   63  0060  86  0D            LDA A  15,I            ; SET "CR" AS NORMAL DELIMITER
   64  0062  D6  00G           LDA B  A.STAT,D        ; IF OUTPUT FUNC.  MODE IS NOT...
   65  0064  2A  20            BPL    3$
   66  0066  D6  00G           LDA B  PPF.JOE,D       ; SET UP DELIMITERS!!!!
   67  0068  27  00            BEQ    1$
   68  006A  B6  0000G         LDA A  PPNUL           ; SET UP NULL CHAR
   69  006D  97  00G           STA A  NULCHR,D
   70  006F  B6  0000G         LDA A  PPEOR
   71  0072  F6  0000G         LDA B  PPEOF
   72  0075  20  13            BRA    2$
   73  0077  D6  00G     1$:   LDA B  A.PRIM,D        ; IF KEYBOARD NO  MODE
   74  0079  C1  1F            CMP B  KBDEV,I
   75  007B  27  09            BEQ    3$
   76  007D  F6  0000G         LDA B  IECRLF          ; CR OR CR/LF FORMAT
   77  0080  27  04            BEQ    3$
   78  0082  86  0A            LDA A  12,I            ; SET LF AS DELIMITER
   79  0084  97  00G           STA A  PPMODE,D        ; ENABLE  MODE
   80  0086  C6  FF      3$:   LDA B  255.,I          ; FF IS NORMAL END OF TEXT
   81  0088  D7  00G           STA B  NULCHR,D        ; SET DEFAULT NULCHR ( MSB ON DISABLES)
   82  008A  97  00G     2$:   STA A  EOLCHR,D
   83  008C  D7  00G           STA B  ETXCHR,D
   84  008E  CE  FFFF          LDX    65535.,I
   85  0091  FF  0000G         STX    POINTB
   86  0094  FE  0000G         LDX    DB
   87  0097  86  AC            LDA A  172.,I
   88  0099  B7  0000G         STA A  SCMSED          ; INITIALIZE SCRAMBLE SEED
   89  009C  4F                CLR A
   90  009D  97  00G           STA A  BLINK,D
   91  009F  97  00G           STA A  NOOUT,D
   92  00A1  97  00G           STA A  CRSTAT,D
   93  00A3  31                INS
   94  00A4  39                RTS
```

```
                                        .SBTTL  BFRALC BUFFER ALLOCATION SCHEME
                                        ;*****************************************
                                        ; THIS ROUTINE SETS UP A.STRT (CHAR. A START OF BUFFER)
                                        ; AND A.MAX (CHAR. A END OF BUFFER).
                                        .GLOBL  MTSREG
00A5  96  00G     BFRALC: LDA A   A.PRIM,D         ; GET DEVICE NUMBER
00A7  81  25              CMP A   37.,I          ; USE IEC BUFFER IF ADDR.>36.
00A9  24  04              BCC     1$
00AB  80  1E              SUB A   30.,I          ; SUBTRACT OUT IEC. ADDRESS
00AD  2E  01              BGT     GTBFRA         ; IF NOT IEC GO GET BUFFER
00AF  4F                  1$:     CLR A          ; SET UP TO GET IEC BUFFER
00B0  CE  00DF'   GTBFRA: LDX     BFRTBL,I       ; GET TABLE ADDRESS
00B3  81  03              CMP A   3,I            ; MAG TAPE?
00B5  26  08              BNE     GTBFR
00B7  D6  00G             LDA B   MTSREG,D       ; IS IT 128 BYTE BUUFER?
00B9  C5  01              BIT B   1,I
00BB  27  02              BEQ     GTBFR
00BD  86  06              LDA A   6,I
00BF  48          GTBFR:  ASL A                  ; MULTIPLY OFFSET BY 4
00C0  48                  ASL A
00C1  16                  TAB                    ; SAVE IT IN B
00C2  BD  0000G           JSR     LDXX           ; GET BUFFER STARTING LOCATION
00C5  DF  00G             STX     A.STRT,D       ; SAVE IT
00C7  DF  00G             STX     A.PTR,D        ; RESET POINTER
00C9  09                  DEX                    ; POINTER INITIALIZATION FOR PUTBYT
00CA  6F  00              CLR     0,X
00CC  DF  00G             STX     A.END,D
00CE  17                  TBA                    ; GET OFFSET BACK

00CF  8B  02              ADD A   2,I            ; MOVE IT 2
00D1  CE  00DF'           LDX     BFRTBL,I       ; TABLE ADDRESS AGAIN
00D4  BD  0000G           JSR     LDXX           ; GET BUFFER ENDING LOCATION
00D7  96  00G             LDA A   A.STAT,D       ; IF INPUT ALLOW 1 CHAR FOR "CR"
00D9  2A  01              BPL     1$
00DB  08                  INX
00DC  DF  00G     1$:     STX     A.MAX,D        ; SAVE IT
00DE  39                  RTS                    ; RETURN
00DF  0000G 0047G BFRTBL: .WORD   IOBFR1, IOBFR1+71.  ; IEC BUFFER
00E3  0000G 0047G         .WORD   EDTBFR, EDTBFR+71.  ; KEYBOARD BUFFER
00E7  0000G 0047G         .WORD   IOBFR1, IOBFR1+71.  ; DISPLAY BUFFER
00EB  0000G 00FFG         .WORD   MTBFR, MTBFR+255.  ; TAPE BUFFER
00EF  0000G 0047G         .WORD   IOBFR1,IOBFR1+71.      ; DATA STATEMENT
00F3  0000G 0047G         .WORD   IOBFR1, IOBFR1+71.  ; MAGTAPE DEVICE 2 (SEPERATE BUFFER)
00F7  0000G 007FG         .WORD   MTBFR,MTBFR+127.       ; SHORT BUFFER
```

```
                                        .SBTTL  UNADR UNADDRESS A DEVICE
0000                            NOIEC   = 0             ; 1 IF NO IEC BUS PINS
                        ;THIS ROUTINE WILL CLEAR THE I/O PROCESSORS STATUS
                        ;       AND UNADDRESS ALL DEVICES
                        ;
                                .GLOBL  ISB
                                .GLOBL  YAXIS
                                .GLOBL  EOLCHR
                                .GLOBL  IECOFF
                                .GLOBL  INTACP
                                .GLOBL  MEMBER
00FB  96  00G     UNADR.        LDA A   ERRCD,D         ; SAVE ERROR CODE WHILE CLEANING UP
00FD  36                        PSH A
00FE  96  00G                   LDA A   IOFUNC,D
0100  D6  00G                   LDA B   A.STAT,D
0102  2A  0F                    BPL     4$
0104  81  00                    CMP A   13.,I           ; IF INPUT THEN SCAN
0106  26  0E                    BNE     3$
0108  96  00G     5$:           LDA A   ERRCD,D         ; GET TO END OF PRESENT LOGICAL
                                                        ; RECORD IF DOING INPUT
010A  9A  00G                   ORA A   CRSTAT,D
010C  26  08                    BNE     3$
010E  BD  0000G                 JSR     MEMBER
0111  20  F5                    BRA     5$
0113  BD  0000G   4$:           JSR     WBYP1           ; PATCH FOR WBY  N: FORM
                                .GLOBL  WBYP1
0116  96  00G     3$:           LDA A   A.PRIM,D        ; IF NOT TALKING TO IEC BUS THEN NO
0118  81  1E                    CMP A   30.,I           ; SEND UNTALK UNLISTEN
011A  22  12                    BHI     UNADR3
011C  BD  0000G                 JSR     ATNON           ; TURN ON ATENTION
011F  BD  0000G                 JSR     INTSRC          ; INITIALIZE IEC AS SOURCE
0122  C6  5F                    LDA B   H5F,I   ; UNTALK
0124  BD  0000G                 JSR     IECSND          ; SEND IT
0127  C6  3F                    LDA B   H3F,I   ; UNLISTEN
0129  BD  0000G                 JSR     IECSND          ; SEND IT
012C  20  08                    BRA     UNFINI
                                .GLOBL  UNADR3  ;NEW ENTRY POINT FOR PATCH XXXXFOR KBPROC BREAK 2 XXXX
012E  81  21      UNADR3:       CMP A   MTPDEV,I
0130  27  04                    BEQ     2$
0132  81  23                    CMP A   MTPD2,I         ; MT DEV. 2
0134  26  03                    BNE     UNFINI
0136  BD  0000G   2$:           JSR     CRTRST          ; RESET THE DISPLAY
0139  BD  0000G   UNFINI:       JSR     IECOFF          ; GET DRIVERS COMPLETELY OFF OF THE BUS
                                .GLOBL  UNADR2          ; MORE PATCH FOR WBY N: FORMAT
013C  4F          UNADR2:       CLR A
013D  97  00G                   STA A   A.STAT,D        ; RESET I/O STATUS
013F  97  00G                   STA A   YAXIS,D
0141  97  00G                   STA A   IOFLGS,D
0143  97  00G                   STA A   PPMODE,D
0145  97  00G                   STA A   NOOUT,D
0147  43                        COM A
0148  97  00G                   STA A   H.PRIM,D
014A  86  0D                    LDA A   15.,I
014C  97  00G                   STA A   EOLCHR,D        ; RESET CR AS EOLCHR
014E  CE  FFFF                  LDX     #7535.,I
0151  FF  0000G                 STX     ISB
0154  32                        PUL A                   ; RESET PROPER ERROR CODE
```

UNADR UNADDRESS A DEVICE

```
 58          0155    97      00G             STA A   ERRCD,D
 59          0157    86      80              LDA A   #80.  ; : CLEAR BREAK AND I/O LIGHT
 60          0159    B7      0000G           STA A   PIALT
 61          015C    39                      RTS
```

```
                                        .SBTTL   .# PROCESSOR
                                        .GLOBL  STKBLD          ; INVERTED STACK BUILDER
                                        .GLOBL  PSHFPN          ; PUSH FPN
                                        .GLOBL  FIX1            ; FIX ROUTINE
                                        .GLOBL  ATPROC          ;    PROCESSOR
                                        .GLOBL  IEC             ;  EVALUATOR CALLED
                                        .GLOBL  FIL             ; #    "
                                        .GLOBL  QS              ; PUTS ON ATSNTG ON STACK FOR LATER
                                        .GLOBL  PT              ;       "
                                        .SBTTL  IEC (  AND PT (  )
                                    ;THESE ROUTINES ARE CALLED FROM THE EVALUATOR AND
                                    ;MEARLY PUSH TAGS AND SET STATUS
                                    ;
0150    23    0000G               PT:   COM   PPMODE          ; SET   MODE
                                    ;
                                    ;xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                                    ; AT CALL TIME IT PUSHES AN ATSNIG
                                    ; AND MARKS  CALLED
                                    ; IN A.STAT
0160                                QS:
0160    32                          IEC:  PUL A                 ; REMOVE RETURN ADDRESS
0161    97    01G                         STA A  DREXTA+1,D
0163    32                                PUL A
0164    97    02G                         STA A  DREXTA+2,D
0166    86    00G                         LDA A  ATSNTG,I       ; ENTER  TAG
0168    36                                PSH A
0169    96    00G                         LDA A  A.STAT,D       ; MARK IT VALID
016B    8A    04                          ORA A  ATVLD,I
016D    97    00G                         STA A  A.STAT,D
016F    7E    0000G                       JMP    DREXTA
```

```
                                        .SBTTL  ATPROC -- OR   SIGN PROCESSOR
                        ATPROC:
0172  86  000G            LDA A   ITM2TG,I       ; SAVE MY EOL TAG
0174  36                  PSH A
0175  9F  000G            STS     R0,D           ; SAVE STACK POINTER
0177  86  000G            LDA A   ATSNTG,I       ; GO LOOK FOR  INFO
0179  BD  0000G           JSR     LOCTG
017C  DE  000G            LDX     R0,D           ; SKIP  TAG
017E  08                  INX
017F  DF  000G            STX     R0,D
0181  BD  0000G           JSR     STKBLD         ; NOW BUILD A REVERSE STACK
0184  33                  PUL B                  ; REMOVE TAG
0185  33                  PUL B                  ; GET FIRST VALUE FROM TYPE ARG
0186  30                  TSX                    ; GET INTER STACK POINTER
0187  EE  00              LDX     0,X
0189  32                  PUL A
018A  32                  PUL A
018B  5D                  TST B
018C  26  45              BNE     ATERR          ; MAKE SURE IT'S A VALUE
018E  08                  INX
018F  BD  0000G           JSR     FIX1           ; FIX IT
0192  26  3F              BNE     ATERR
0194  A6  03              LDA A   FIX1A,X        ; GET MSB OF FIXED NO
0196  26  38              BNE     ATERR
0198  A6  04              LDA A   FIX1B,X
019A  27  37              BEQ     ATERR
019C  81  23              CMP A   35.,I
019E  27  33              BEQ     ATERR
01A0  97  00G             STA A   A.PRIM,D       ; SAVE IT
01A2  96  00G             LDA A   A.STAT,D       ; MARK IT DEFINED
01A4  8A  01              ORA A   PRIDEF,I
01A6  84  F8              AND A   #DEFB,I ; TURN OFF ATVLD -MCD- !!!!!!!!!!
01A8  97  00G             STA A   A.STAT,D
01AA  33                  PUL B
01AB  C1  00G             CMP B   ITM2TG,I       ; IF MY EOL THEN EXIT
01AD  26  01              BNE     ATSEC
01AF  39                  RTS
01B0  33            ATSEC:PUL B                  ; TYPE
01B1  30                  TSX                    ; GET INTER STACK POINTER
01B2  EE  00              LDX     0,X
01B4  32                  PUL A
01B5  32                  PUL A
01B6  5D                  TST B
01B7  26  1A              BNE     ATERR
01B9  08                  INX
01BA  BD  0000G           JSR     FIX1           ; FIX IT
01BD  26  14              BNE     ATERR
01BF  A6  03              LDA A   FIX1A,X        ; IS MSB=0
01C1  26  10              BNE     ATERR
01C3  A6  04              LDA A   FIX1B,X        ; GET VALID 8 BITS
01C5  81  21              CMP A   33.,I          ; SEC.<33 ?
01C7  24  0A              BCC     ATERR
01C9  97  00G             STA A   A.SEC,D        ; SAVE SECONDARY ADDRESS
01CB  96  00G             LDA A   A.STAT,D
01CD  8A  02              ORA A   SECDEF,I
01CF  97  00G             STA A   A.STAT,D
01D1  33                  PUL B                  ; REMOVE MY EOL TAG
```

```
    58          0102    39                              RTS                     ; EXIT
    59          0103    86      00G             AIERR:  LDA A   ERRTSN,I        ; DEVICE ADDRESS
    60          0105    97      00G                     STA A   ERRCD,D         ; ARGUMENT ERROR
    61          0107    86      00G                     LDA A   ITM2TG,I        ; FIND MY RETURN ADDRESS
    62          0109    9F      00G                     STS     R0,D            ; SAVE USP
    63          010B    AD      0000G                   JSR     LOCTG
    64          010E    9E      00G                     LDS     R0,D
    65          01E0    32                              PUL A
    66          01E1    39                              RTS                     ; EXIT
```

```
                        .SBTTL  FIL----- # SIGN PROCESSOR
                ; THIS ROUTINE STORES THE LOGICAL UNIT NO. IN LUNNO
                ; AND STORES THE RECORD NUMBER IN RECNO IF ONE EXISTS.
                ;
                        .GLOBL  LUNNO           ; PRESENT LUN NO.
                        .GLOBL  RECNO           ; PRESENT REC NO.
                ;
                        .IF NDF FILSYS
                        .ENDC
```

```
                                        .SBTTL  INAITT--INTEGER TO ASCII
                                ;THIS ROUTINE CONVERTS AN INTEGER TO AN ASCII STRING
                                ;INPUT: R0 IS INTEGER
                                ;       R1 TELLS WHERE TO PUT ASCII STRING
                                ;       R2 IS MAX LENGTH
                                ;
                                        .GLOBL  INAITT          ; INTEGER --> ASCII
                                        .GLOBL  FPAITT          ; FLOATING POINT --> ASCII
                                        .GLOBL  FLOAT1
                                        .GLOBL  PRIVAL
                                        .GLOBL  INPVAL
01E2    96      01G     INAITT: LDA A   R0+1,D          ; PUT NUMBER ON STACK
01E4    36                      PSH A
01E5    96      00G             LDA A   R0,D
01E7    36                      PSH A
01E8    36                      PSH A                   ; THAT'S A TAG?
01E9    BD      0000G           JSR     FLOAT1          ; FLOAT IT
01EC    CE      0000G           LDX     FPC,I
01EF    BD      0000G           JSR     PULFPN
01F2    CE      0000G   FPAITT: LDX     FPC,I           ; TRANSFER IT TO FPC
01F5    DF      00G             STX     POINT,D
01F7    CE      0000G           LDX     A.STAT,I        ; SAVE PRESENT I/O STATUS
01FA    BD      0000G           JSR     PSHFPN
01FD    96      00G             LDA A   R1,D
01FF    D6      01G             LDA B   R1+1,D          ;GET ORIGIN
0201    DE      00G             LDX     R2,D            ; SUBTRACT ONE FROM THE LENGTH
0203    09                      DEX
0204    DF      00G             STX     R2,D
0206    DB      01G             ADD B   R2+1,D          ;ADD OUTPUT ORIGIN
0208    99      00G             ADC A   R2,D
020A    97      00G             STA A   A.MAX,D         ; SAVE IT FOR I/O DRIVER
020C    D7      01G             STA B   A.MAX+1,D
020E    DE      00G             LDX     R1,D            ; SET UP POINTER FOR A CHANEL
0210    DF      00G             STX     A.STRT,D        ; THIS SO CAN FIND IT LATER
0212    09                      DEX                     ; GET POINTER TO REAL PLACE
0213    DF      00G             STX     A.END,D
0215    6F      00              CLR     0,X             ; MAKE SURE THERE ISN'T ONE OF THOSE
                                                        ; LITTLE "CR" IN THE BUFFER!!!!!!
0217    7F      0000G           CLR     A.STAT          ; CLEART STATUS
021A    86      01              LDA A   NOWRIT,I        ; NO TRANSFERS JUST SET ERRCD
021C    97      00G             STA A   NOOUT,D         ; IF OVERFLOW STRING
021E    86      24              LDA A   ITTDEV,I        ; SET UP FOR INTERNAL XFER
0220    97      00G             STA A   A.PRIM,D
0222    86      02              LDA A   2,I             ; SET SEMI FLAG SO IT WON'T TAB
0224    97      00G             STA A   IOFLGS,D
0226    BD      0000G           JSR     PRIVAL          ; OUTPUT ASCII
0229    7E      0000G           CLR     NOOUT           ; CLEAN UP THE MESS YOU MADE
022C    DE      00G             LDX     A.END,D         ; PUT POINTERS BACK
022E    DF      00G             STX     R2,D
0230    DE      00G             LDX     A.STRT,D        ; GET POINTER TO FIRST VALID CHAR.
0232    DF      00G             STX     R1,D
0234    CE      0000G           LDX     A.STAT,I        ; RESTORE STATUS
0237    BD      0000G           JSR     PULFPN
023A    39                      RTS
```

```
                                        .SBTTL  AFPITT--ASCII TO FLOATING POINT INTERNAL
                                ;THIS ROUTINE DOES AN INTERNAL INPUT OPERATION
                                ;ENTRY: R0 IS START OF TEXT
                                ;       R1 IS END OF TEXT +1
                                ;RETURN:        FPC (PAGE N) IS FLOATING POINT NUMBER
                                ;       R0 IS BREAK CHARACTER
                                ;       R1 IS END OF TEXT +1
                                ;
                                        .GLOBL  AFPITT
0238    CE      00000G  AFPITT: LDX     A.STAT,I        ; PUSH OLD STATUS
023E    BD      00000G          JSR     PSHFPN
0241    86      A0              LDA A   BFRSTT!DIRCT,I  ; SET STATUS
0243    97      00G             STA A   A.STAT,D
0245    DE      00G             LDX     R0,D    ; SET TO START INPUT
0247    DF      00G             STX     A.PTR,D
0249    DF      00G             STX     A.STRT,D
0248    DE      00G             LDX     R1,D
024D    DF      00G             STX     A.END,D         ; SET UP MAXIMUM SIZE
024F    CE      00000G          LDX     FPC,I           ; SET RESULT STORAGE AREA
0252    DF      00G             STX     POINT,D
0254    86      24              LDA A   ITTDEV,I
0256    97      00G             STA A   A.PRIM,D
0258    7F      00000G          CLR     CRSTAT
0258    BD      00000G          JSR     INPVAL          ; INPUT VALUE
025E    DE      00G             LDX     A.PTR,D         ; SET UP R0 TO NEW POSITION
0260    DF      00G             STX     R0,D
0262    DE      00G             LDX     A.END,D         ; RESTORE R1
0264    DF      00G             STX     R1,D
0266    CE      00000G          LDX     A.STAT,I        ; RESTORE STATUS
0269    BD      00000G          JSR     PULFPN
026C    39                      RTS
```

```
                                                .SBTTL  FIXNUM--FIXES A PRINT ENTRY
                                        ;THIS ROUTINE FIXES A NUMBER POINTER TO BY THE I/O PROCESSOR VARIABLE POINT
                                        ;IF IT IS A STRING THE STRING WILL BE CONVERTED.  AT RETURN TIME
                                        ;THE FLOATING VALUE WILL BE IN FPC AND THE INTEGER WILL
                                        ;BE IN THE X REG
                                        ;ENTRY: DT = 0   IF VALUE       NORMALLY SET BY IOPROC
                                        ;           1  IF STRING
                                        ;       POINT = POINTER TO FPN
                                        ;               OR STRING
                                        ;EXIT:  FPC <-- FLOATING NUMBER
                                        ;       X REG  <-- INTEGER
                                                .GLOBL  FIXNUM
                                                .GLOBL  FPB
                                                .GLOBL  ERINTR
0260  96  00G     FIXNUM: LDA A   DT,D             ; IS IT A VALUE INPUT?
0262  4A                  DEC A
0263  27  06              BEQ     FIXVAL
0265  86  00G             LDA A   ERINTR,I         ; NO STRINGS FOR NOW
0267  97  00G             STA A   ERRCD,D
0269  0A                  SEV                     ;SET OVERFLOW BIT AS FLAG
026A  39                  RTS                     ;EXIT.....
026B  DE  00G     FIXVAL: LDX     POINT,D         ; GET NUMBER
026D  BD  0000G           JSR     PSHFPN
0270  BD  0000G           JSR     PSHFPN          ; COPY IT
0273  CE  0000G           LDX     FPC,I           ; PUT IT WHERE IT SHOULD BE
0276  BD  0000G           JSR     PULFPN
0279  CE  0000G           LDX     FPB,I
027C  BD  0000G           JSR     PULFPN
027F  09                  DEX
0280  BD  0000G           JSR     FIX1            ; FIX THE ONE IN FPB
0283  5F                  CLR B
0284  4D                  TST A
0285  27  02              BEQ     1$
0287  0A                  SEV                     ;SET EVERFLOW BIT AS FLAG
0288  39                  RTS
0289  E6  04      1$:     LDA B   FIX1B,X
028B  A6  03              LDA A   FIX1A,X
028D  27  01              BEQ     2$
028F  0D                  SEC                     ;SET CARRY BIT AS FLAG
0290  EE  03      2$:     LDX     FIX1A,X
0292  39                  RTS
```

FIXNUM--FIXES A PRINT ENTRY

```
                                        .GLOBL  INAGTE
                                    ;CONVERTS AN INTEGER TO ASCII.
                                    ;
                                    ;PASS IN X THE NUMBER TO BE CONVERTED.
                                    ;RESULT STRING IS IN R15, R16, R17, R18, R19, AND R20 WITH A NULL AFTER
                                    ;THE LAST CHARACTER.

02A0    DF      00G         INAGTE. STX     R0,0
02A2    CE      0000G               LDX     R16,1
02A5    DF      00G                 STX     R1,0
02A7    CE      0008                LDX     8.,1
02AA    DF      00G                 STX     R2,0
02AC    BD      01E2'               JSR     INAITT
02AF    DE      00G                 LDX     R2,0
02B1    6F      01                  CLR     1,X
02B3    39                          RTS                     ;BYE
                0001'               .END
```

SYMBOL TABLE

```
ABRFLG=  0040        ADRDEV   0000RG  02  ADRDON   1050R   02  ADRD.    0058R   02  ADRMOR   0048R   02
ADROK    0010R   02  AFAIL =  0030        AFPITT   023BRG  02  AMAT =   0010        ARRAY =  0020
AS       0160RG  02  ASTR =   0020        ATERR    01D3R   02  ATLOD =  0000        ATNON =  XXXXXX G
ATPROC   0172RG  02  ATSEC    0180R   02  ATSNTG=  XXXXXX G    ATVLD =  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    BAKSTG=  XXXXXX G    BANK  =  XXXXXX G    BFRALC   0040RG  02  BFRSTT=  0020
BFRTBL   000FR   02  BLINK =  XXXXXX G    BMAT  =  0004        BRKCNT=  XXXXXX G    BSTR  =  0008
BUSACT=  0010        CDOPTR=  XXXXXX G    COSPTR=  XXXXXX G    CHAR  =  XXXXXX G    CHRCNT=  XXXXXX G

CLPTR =  XXXXXX G    CMAT  =  0001        COLCNT=  XXXXXX G    CRDC3 =  0002        CREOF =  0008
CREOI =  0004        CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT=  XXXXXX G
CRTRST=  XXXXXX G    CRVLD =  0080        CSTR  =  0002        CTKN  =  XXXXXX G    CURSOR=  XXXXXX G

DATDEV=  0022        DB    =  XXXXXX G    DIMFLG=  0004        DIRCT =  0080        DISCNT=  XXXXXX G
DISSRQ=  0008        DL    =  XXXXXX G    DNADDR   0031R   02  DP    =  XXXXXX G    DREXTA=  XXXXXX G
DREXTB=  XXXXXX G    DSPVEC=  0020        DSPSTT=  XXXXXX G    DT    =  XXXXXX G    EDTBFR=  XXXXXX G

ENDKEY=  0040        EOFTYP=  0038        EOLCHR=  XXXXXX G    EOLTG =  XXXXXX G    EOSTG =  XXXXXX G
ERATSN=  XXXXXX G    ERJOMN=  XXXXXX G    EREOM =  XXXXXX G    ERFBFR=  XXXXXX G    ERFILE=  XXXXXX G
ERINTR=  XXXXXX G    ERIOE =  XXXXXX G    ERNEST=  XXXXXX G    ERNLOD=  XXXXXX G    ERNSEP=  XXXXXX G

ERRCD =  XXXXXX G    ERTERM=  XXXXXX G    ERUNOF=  XXXXXX G    ESTG  =  XXXXXX G    ETXCHR=  XXXXXX G
EXTFLG=  0080        FIL   =  XXXXXX G    FILDEV=  0000        FIXNUM   0260RG  02  FIXVAL   0278R   02
FIX1  =  XXXXXX G    FIX10 =  0003        FIX18 =  0004        FLOAT1=  XXXXXX G    FMTWLD=  0008

FNFLG =  0010        FORTG =  XXXXXX G    FPAITT   01F2RG  02  FPB   =  XXXXXX G    FPC   =  XXXXXX G
GLBFLG=  XXXXXX G    GOSTG =  XXXXXX G    GTBFR    008FR   02  GTBFRA   00B0R   02  IEC      0160RG  02
IECOFF=  XXXXXX G    IECRLF=  XXXXXX G    IECSND=  XXXXXX G    INDXIG=  XXXXXX G    INAGTE   0290RG  02

INAITT   01E2RG  02  INPUTK=  0001        INPVAL=  XXXXXX G    INTACP=  XXXXXX G    INTSRC=  XXXXXX G
IOBFR1=  XXXXXX G    IOFLGS=  XXXXXX G    IOFUNC=  XXXXXX G    IOPLT =  0040        IOSYSF=  XXXXXX G
ISR   =  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    L9RKTG=  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        LUNNO =  XXXXXX G    MAGEND=  XXXXXX G    MTBFR =  XXXXXX G
MTPADR=  XXXXXX G    MTPDEV=  0021        MTPD2 =  0023        MTSREG=  XXXXXX G    NEWBFR=  XXXXXX G
NLPTR =  XXXXXX G    NOIEC =  0000        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

OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002
OPRADR=  XXXXXX G    PAETG =  XXXXXX G    PARM  =  0008        PGMATR=  0002        PGMSP =  0005
PGMCD =  0009        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    PLOSTG=  XXXXXX G    PNDEOF=  XXXXXX G    PNDFLG=  XXXXXX G    PNTNTG=  XXXXXX G
PNTSTG=  XXXXXX G    POINT =  XXXXXX G    POINTB=  XXXXXX G    PPEOF =  XXXXXX G    PPEOR =  XXXXXX G

PPMODE=  XXXXXX G    PPNUL =  XXXXXX G    PRI      0027R   02  PRIDEF=  0001        PRIVAL=  XXXXXX G
PRTTG =  XXXXXX G    PSCTG =  XXXXXX G    PSHFPN=  XXXXXX G    PT       0150RG  02  PULFPN=  XXXXXX G
PUPTAP=  XXXXXX G    RECLFG=  0004        RECNO =  XXXXXX G    REAIL =  00C0        RMAT  =  0040

RPTCTL=  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    SCRTCH=  XXXXXX G    SECDEF=  0002        SEMITG=  XXXXXX G
SNOIT =  0080        STAT37=  XXXXXX G    STKBLD=  XXXXXX G    STPFLG=  0040        STPKEY=  0020
STRING=  0010        SYSERR=  XXXXXX G    TABCNT=  XXXXXX G    TABPTR=  XXXXXX G    TAGPTR=  XXXXXX G

TCOL  =  XXXXXX G    TRCFLG=  0020        T510EV=  0025        UNADR    00FBRG  02  UNADR2   013CRG  02
UNADR3   012ERG  02  UNDEF =  0080        UNFINI   0139R   02  VALTG =  XXXXXX G    VALUND=  0040
WBYP1 =  XXXXXX G    XAXIS =  XXXXXX G    YAXIS =  XXXXXX G

. ABS.   0000    00
```

```
IOCTL  I/O CONTROL ROUTINES    RT-11 MMAC  VM02-10   14-OCT-76 01:34:51 PAGE 11+
SYMBOL TABLE

        0000     01
IOCTL   0284     02

ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2297. WORDS
.SY:IOCTL/C<DK1:SEICL1,IOCTL
```

CROSS REFERENCE TABLE (CREF V01-03)

```
A.END    1-196#  3-27   8-36   8-48    9-18   9-27
A.MAX    1-197#  3-36   8-31   8-32
A.PRIM   1-192#  2-40   2-48   2-73    3-6    4-27   4-52   6-29   8-43   9-22
A.PTR    1-194#  3-24   9-15   9-25
A.SEC    1-193#  2-35   6-53
A.STAT   1-191#  2-20   2-24   2-30    2-47   2-64   3-33   4-15   4-46   5-27   5-29   6-30   6-33   6-54
         6-56    8-22   8-39   8-52    9-10   9-13   9-29
A.STRT   1-195#  3-23   8-34   8-50    9-16
ABRFLG   1-26#
ADRD     2-54    2-56   2-60#
ADRDEV   1-278#  2-16#
ADRDON   2-52    2-57#
ADRMOR   2-50    2-53#
ADROK    2-23    2-31#
AFAIL    1-155#
AFPITT   9-9#    9-10#
AMAT     1-154#
ARRAY    1-83#
AS       5-8#    5-20#
ASTR     1-153#
ATERR    6-19    6-22   6-24   6-26    6-28   6-44   6-47   6-49   6-52   6-59#
ATLOO    1-216#
ATNON    1-283#  4-30
ATPROC   1-282#  2-25   5-5#   6-2#
ATSEC    6-36    6-38#
ATSNTG   1-138#  5-25   6-6
ATVLD    1-204#  2-22   5-28
BAKSTG   1-146#
BANK     1-250#
BFRALC   1-279#  3-6#
BFRSTT   1-201#  2-21   9-12
BFRTBL   3-12    3-31   3-38#
BLINK    1-222#  2-90
BMAT     1-157#
BRKCNT   1-73#
BSTR     1-156#
BUSACT   1-202#  2-21
CDOPTR   1-71#
CDSPTR   1-70#
CHAR     1-82#
CHRCNT   1-242#
CLPTR    1-20#
CMAT     1-159#
COLCNT   1-244#
CRDC3    1-181#
CREOF    1-183#
CREOI    1-182#
CREOT    1-185#
CRETX    1-184#
CRNORM   1-180#
CRSTAT   1-178#  2-92   4-21   9-23
CRTRST   1-291#  4-42
CRVLD    1-187#
CSTR     1-158#
CTKN     1-27#
CURSOR   1-223#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
DATDEV  1-258#  2-55
DB      2-6#    2-86*
DIMFLG  1-28#
DIRCT   1-200#  9-12
DISCNT  1-221#
DISSRQ  1-36#
DL      1-233#
DNADDR  2-37    2-41#
DP      1-232#
DREXTA  1-271#  5-22*  5-24*  5-30
DREXTB  1-272#
DSPDEV  1-256#  2-53
DSPSTT  1-287#
DT      1-234#  10-15
EOTBFR  1-265#  3-39    3-39
ENDKEY  1-213#
EOFTYP  1-186#
EOLCHR  2-11#   2-82*   4-8#    4-54*
EOLTG   1-142#
EOSTG   1-143#
ERATSN  1-170#  6-59
ERDOMN  1-164#
EREOM   1-171#
ERFBFR  1-167#
ERFILE  1-169#
ERINTR  10-14#  10-18
ERIOE   1-168#
ERNEST  1-295#
ERNIOD  1-173#
ERNSEP  1-165#
ERRCO   1-42#   2-26    4-12    4-19    4-58*   6-60*   10-19*
ERTERM  1-166#
ERUNDF  1-172#
ESTG    1-124#
ETXCHR  2-11#   2-83*
EXTFLG  1-25#
FIL     5-7#
FILDEV  1-254#
FILSYS  7-8     7-11    7-74
FIX1    5-4#    6-21    6-46    10-30
FIX1A   1-117#  6-23    6-48    10-37   10-40
FIX1B   1-118#  6-25    6-50    10-36
FIXNUM  10-12#  10-15#
FIXVAL  10-17   10-22#
FLOAT1  8-9#    8-17
FMTVLD  1-207#
FNFLG   1-27#
FORTG   1-126#
FPAITT  8-8#    8-20#
FPB     10-13#  10-27
FPC     1-292#  8-18    8-20    9-19    10-25
GLBFLG  1-31#
GOSTG   1-125#
GTBFR   3-14    3-17    3-19#
GTBFRA  3-10    3-12#
IEC     5-6#    5-21#
IECOFF  4-9#    4-43
```

CROSS REFERENCE TABLE (CREF V01-03)

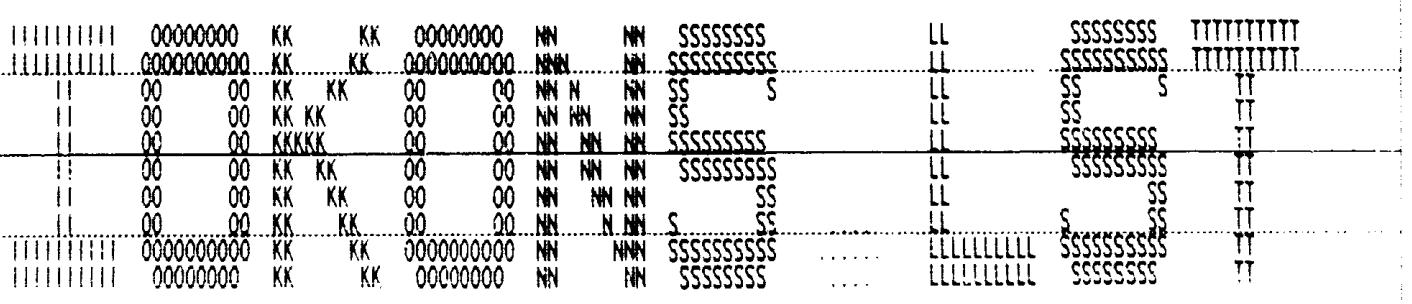
CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
R17     1-39#
R18     1-39#
R19     1-39#
R2      1-38#   8-26    8-28*   8-29    8-30    8-49*   11-14*  11-16
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#
R9      1-38#
RECLEG  1-217#
RECNO   7-6#
RFAIL   1-152#
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNTG  1-140#  2-16
RUNFLG  1-32#
RUNN    1-218#
SAP     1-47#
SCALER  1-82#
SCMSED  2-9#    2-88*
SCRTCH  1-277#
SECDEF  1-205#  2-31    2-46    6-55
SEMITG  1-144#
SNDIT   1-228#
STAT37  1-247#
STKBLD  1-286#  5-2#    6-11
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSE.R  1-43#
T51DEV  1-261#
TABCNT  2-12#   2-61*
TABPTR  1-243#
TAGPTR  1-239#
TCOL    1-245#
TRCFLG  1-34#
UNADR   1-280#  4-12#
UNADR2  4-44#   4-45#
UNADR3  4-29    4-37#   4-38#
UNDEF   1-81#
UNFINI  4-36    4-41    4-43#
VALTG   1-134#
VALUNO  1-88#
WBYP1   4-25    4-26#
XAXIS   1-248#
YAXIS   4-7#    4-47*
```

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3#



```
                                .TITLE  IOKONS I/O CONSTANTS
                                .ENABL HEX
                                .GLOBL  IOKONS
0000'                           IOKONS  =.
                                .IDENT  /SB1009/
                        ;THESE DEFINE A TABLE IN THE EVALUATOR SO THAT THE REST OF THE I/O
                        ;       PACKAGE CAN BE SIMLIFIED.
                        ;FORMAT OF EACH 2 BYTE ENTRY:
                        ;       BYTE 1  BIT 7   STATUS BIT      . PRESENTLY USED BY GRAPHICS
                        ;                                       ; TO INDICATE RELATIVE MODE
                        ;               BIT 6   UNUSED
                        ;               BIT 5-0 DEFAULT PRIMARY ADDRESS
                        ;
                        ;       BYTE 2  BIT 7   DIRECTION P.I.  ; SET IF INPUT
                        ;               BIT 6   UNUSED
                        ;               BIT 5-0 DEFAULT SECONDARY ADDRESS
                        ;
                        ;A MACRO TO EASE THIS MESS
000A                            .RADIX 10
                                .LIST   ME,MC,MD,SEQ
                        .NLIST  TTM,LD,LOC
                        .MACRO  IO      ID,SA,DIR,PA
                        .NLIST
                        K1=SA*256
                        K2=DIR*128+PA
                        .GLOBL ID
                        .LIST
                   ID   = K1+K2
                        .ENDM
                                IO      SAVE,1,0,33
0121               SAVE  = K1+K2
                                IO      OPEN,3,0,0
0300               OPEN  = K1+K2
                                IO      OLD,4,1,33
04A1               OLD   = K1+K2
                                IO      CREATE,5,0,0
0500               CREATE= K1+K2
                                IO      KILTYP,7,0,33
0721               KILTYP= K1+K2
                                IO      UNIT,8,0,0
0800               UNIT  = K1+K2
                                IO      DIR,9,0,0
0900               DIR   = K1+K2
                                IO      COPTYP,10,0,32
0A20               COPTYP= K1+K2
                                IO      RELBL,11,0,0
0800               RELBL = K1+K2
                                IO      PRINT,12,0,32
0C20               PRINT = K1+K2
                                IO      INPUT,13,1,31
009F               INPUT = K1+K2
                                IO      READ,14,1,34
0EA2               READ  = K1+K2
                                IO      WRITE,15,0,33
0F21               WRITE = K1+K2
                                IO      ASSIGN,16,0,0
1000               ASSIGN= K1+K2
```

```
                                        .IO     LIST,19,0,32
1320            LIST    = K1+K2
                                        .IO     ZDRAW,20,0,32
1420            ZDRAW   = K1+K2
                                        .IO     ZRDRAW,128+20,0,32
9420            ZRDRAW  = K1+K2
                                        .IO     ZMOVE,21,0,32
1520            ZMOVE   = K1+K2
                                        .IO     ZRMOVE,128+21,0,32
9520            ZRMOVE  = K1+K2
                                        .IO     PAGE,22,0,32
1620            PAGE    = K1+K2
                                        .IO     HOME,23,0,32
1720            HOME    = K1+K2
                                        .IO     ZGIN,24,1,32
18A0            ZGIN    = K1+K2
                                        .IO     CMD,26,0,0
1A00            CMD     = K1+K2
                                        .IO     MARK,28,0,33
1C21            MARK    = K1+K2
                                        .IO     FIND,27,0,33
1B21            FIND    = K1+K2
                                        .IO     SECFNC,29,0,37
1D25            SECFNC  = K1+K2
                                        .IO     SPCIOF,16,0,32  ; SPECIAL I/O FUNCTIONS THAT NEED TO LOOK
1020            SPCIOF  = K1+K2
                                                                ; LIKE A PRINT USE THIS
        0001'                           .END
```

```
IOKONS I/O CONSTANTS   RT-11 MMAC  VM02-10   14-OCT-76 01:35:09 PAGE 1+
SYMBOL TABLE

ASSIGN=  1000 G     CMD    =  1A00 G     COPTYP=  0A20 G     CREATE=  0500 G     DIR    =  0900 G
FIND   =  1821 G    HOME   =  1720 G     INPUT  =  009E G     IOKONS=  0000RG     KILTYP=  0721 G
K1     =  1000      K2     =  0020       LIST   =  1320 G     MARK   =  1C21 G     OLD    =  04A1 G
OPEN   =  0300 G    PAGE   =  1620 G     PRINT  =  0C20 G     READ   =  0EA2 G     RELBL  =  0800 G
SAVE   =  0121 G    SECENC=  1025 G      SPCIOF=  1020 G     UNIT   =  0800 G     WRITE  =  0F21 G
ZDRAW  =  1420 G    ZGIN   =  18A0 G     ZMOVE  =  1520 G     ZRDRAW=  9420 G     ZRMOVE=  9520 G

  .BBS.   0000      00
          0000      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3293.  WORDS
.SY:IOKONS/C(DK1:SETCLI.IOKONS
```

CROSS REFERENCE TABLE (CREF V01-03)

	1-19													
ASSIGN	1-58#	1-58#												
CMD	1-67#	1-67#												
CQPTYP	1-52#	1-52#												
CREATE	1-48#	1-48#												
DIR	1-51#	1-51#												
FIND	1-69#	1-69#												
HOME	1-65#	1-65#												
INPUT	1-55#	1-55#												
IOKONS	1-18#	1-19#												
K1	1-45	1-45#	1-46	1-46#	1-47	1-47#	1-48	1-48#	1-49	1-49#	1-50	1-50#	1-51	1-51#
	1-52	1-52#	1-53	1-53#	1-54	1-54#	1-55	1-55#	1-56	1-56#	1-57	1-57#	1-58	1-58#
	1-59	1-59#	1-60	1-60#	1-61	1-61#	1-62	1-62#	1-63	1-63#	1-64	1-64#	1-65	1-65#
	1-66	1-66#	1-67	1-67#	1-68	1-68#	1-69	1-69#	1-70	1-70#	1-71	1-71#		
K2	1-45	1-45#	1-46	1-46#	1-47	1-47#	1-48	1-48#	1-49	1-49#	1-50	1-50#	1-51	1-51#
	1-52	1-52#	1-53	1-53#	1-54	1-54#	1-55	1-55#	1-56	1-56#	1-57	1-57#	1-58	1-58#
	1-59	1-59#	1-60	1-60#	1-61	1-61#	1-62	1-62#	1-63	1-63#	1-64	1-64#	1-65	1-65#
	1-66	1-66#	1-67	1-67#	1-68	1-68#	1-69	1-69#	1-70	1-70#	1-71	1-71#		
KILTYP	1-49#	1-49#												
LIST	1-59#	1-59#												
MARK	1-68#	1-68#												
OLD	1-47#	1-47#												
OPEN	1-46#	1-46#												
PAGE	1-64#	1-64#												
PRINT	1-54#	1-54#												
READ	1-56#	1-56#												
RELBL	1-53#	1-53#												
SAVE	1-45#	1-45#												
SECFNC	1-70#	1-70#												
SPCIOF	1-71#	1-71#												
UNIT	1-50#	1-50#												
WRITE	1-57#	1-57#												
ZDRAW	1-60#	1-60#												
ZGIN	1-66#	1-66#												
ZMOVE	1-62#	1-62#												
ZRDRAW	1-61#	1-61#												
ZRMOVE	1-63#	1-63#												

CROSS REFERENCE TABLE (CREF V01-03)

IO	1-37#	1-45	1-46	1-47	1-48	1-49	1-50	1-51	1-52	1-53	1-54	1-55	1-56	1-57
	1-58	1-59	1-60	1-61	1-62	1-63	1-64	1-65	1-66	1-67	1-68	1-69	1-70	1-71
SEI	1-3#													

TABLE OF CONTENTS

```
                        .TITLE  IOPROC  I/O PROCESSOR
                        .IDENT  /SBGD42/
                        .GLOBL  IOPROC
                        .GLOBL  IOSCAN
                        .GLOBL  IOCLNR
                .SBTTL  GOOD OLD GLOBALS AND ASSIGNMENTS


                ;       ******** LOOK AT UPDLEN FOR GEORGE'S CHANGES **********


                ;
                ; GLOBAL SUBROUTINES
                .GLOBL INPVAL,WRIVAL,PRIVAL,REAVAL,RBYTE,WBYTE
                .GLOBL INPSTG,WRISTG,PRISTG,REASTG
                        .GLOBL  DREXTA          ; GOOD OLD DYNAMIC CODE
                ; GLOBAL GOODIES
                        .GLOBL  ADRDEV          ; SETS UP DEVICE ADDRESSES
                        .GLOBL  BFRALC          ; BUFFER ALLOCATION ROUTINE
                        .GLOBL  BACKUP          ; TO BACK UP THE STACK
                        .GLOBL  CRLFLF          ; PRINT A BLANK LINE !!!!
                        .GLOBL  INT1            ; FOR STRING DIMENSION
                        .GLOBL  DIMSTR          ; STRING DIMENSION
                        .GLOBL  ASX             ; MOVE THE INDEX REG
                        .GLOBL  ABX
                        .GLOBL  SNDBFR          ; SEND CURRENT BUFFER
                        .GLOBL  PUTBYT          ; SEND ONE CHARACTER TO BUFFER
                        .GLOBL  UNADR           ; UNADDRESS CURRENT DEVICES
                        .GLOBL  FMTINI          ; INITIALIZE FORMAT
                        .GLOBL  FMTCLN          ; FORMAT CLEANUP
                        .GLOBL  ISB             ; IMAGE STRING BASE POINTER
                        .GLOBL  ISP             ; EVAL'S IMAGE POINTER
                ;
                ; LOCAL FLAGS, POINTERS
                ;       TAGPTR          ; POINTER TO PRESENT TAG IN I/O LIST
                ;       LENGTH          ; LENGTH WORD
                ;       POINT           ; POINTER TO STRING/VALUE
                ;       COLCNT          ; NUMBER OF DIMENSIONED COLS.
                ;       TCOL            ; TEMP OF COLCNT
                ;       IOFLGS          ; I/O PROCESSOR STATUS FLAGS
0001                    MTR     = H01   ; BIT 0 - MATRIX
0002                    SEMIC   = H02   ; BIT 1 - SEMICOLON
0008                    STGFLG  = H08   ; BIT 3 - PROCESSING A STRING
0010                    ENDIO   = H10   ; BIT 4 - END I/O PREMATURELY
0020                    LASTYP  = H20   ; BIT 5 - LAST TYPE OF I/O 0=VALUE
                                        ;                          1=STRING
                ;       IOSNTP          ; POINTER TO NAME TABLE ENTRY
                ;       CHRCNT          ; ACTUAL STRING LENGTH INPUT
```

```
                                .SBTTL  IOPROC--MAINLINE CONTROL FOR THE I/O LIST PROCESSOR
                              ;THIS ROUTINE DOES THE NORMAL CONTROL FOR THE I/O LIST PROCESSOR
                              ;
0000   86   000G      IOPROC: LDA A   EOLTG,I         ; TAG IT SO I CAN FIND IT
0002   36                    PSH A
0003   9F   000G              STS     R0,0            ; SET UP VSP
0005   BD   092A'            JSR     IOSCAN          ; GO DO I/O
0008   BD   000B'            JSR     IOCLNR          ; CLEAN UP AND EXIT
```

```
                                .SBTTL  IOCLNR--I/O CLEAN UP ROUTINE
                        ;THIS ROUTINE SERCHES FOR THE FIRST OCCURANCE OF THE "EOLTG" AND
                        ;USES THE ASSOCIATED INFORMATION AS THE RETURN ADDRESS.  IT THEN
                        ;SEARCHES FOR A SECOND "EOLTG" AND MOVES THE STACK POINTER THERE
                        ;THEREBY CLEANING UP THE PRINT LIST ON THE STACK.  AFTER ALL OF
                        ;THIS CONTROL IS PASSED TO THE RETURN ADDRESS RETRIEVED EARLIER
                        ;
0008  86  00G     IOCLNR: LDA A   RTRNTG,I        ; STACK A TAG
0000  36                  PSH A
000E  9F  00G             STS     R0,D            ; SAVE STACK POINTER
0010  86  00G             LDA A   EOLTG,I         ; FIND FIRST EOLTG
0012  BD  0000G           JSR     LOCTG
0015  9E  00G             LDS     R0,D            ; MOVE STACK THERE
0017  32                  PUL A
0018  32                  PUL A                   ; SAVE RETURN ADDRESS
0019  97  01G             STA A   DREXTA+1,D
0018  32                  PUL A
001C  97  02G             STA A   DREXTA+2,D
001E  9F  00G             STS     R0,D            ; UPDATE VSP
0020  86  00G             LDA A   EOLTG,I         ; FIND SECOND EOLTG
0022  BD  0000G           JSR     LOCTG
0025  9E  00G             LDS     R0,D
0027  7E  0000G           JMP     DREXTA          ; RETURN TO ORIGINAL
```

```
                                .SBTTL  I/O PRE-PROCESSOR
                                ;*****************************************
                                ; THIS SECTION PREFORMS THE FOLLOWING INITIALIZATION:
                                ; 1) SET UP DEVICE ADDRESSES FOR I/O
                                ; 2) SET UP APPROPRIATE BUFFERS
                                ; 3) INITIALIZE THE I/O LIST STACK - THIS CONSISTS IN
                                ;    BACKING UP THE STACK AND PUSHES THE ADDRESS OF EACH
                                ;    TAGGED ITEM IN THE LIST.  THIS CONTINUES UNTIL THE END
                                ;    OF LINE TAG IS FOUND.
002A  86  000   IOSCAN: LDA A   #ITM2TG,I        ; PUT AN "EOL" TAG ON THE STACK FOR ME!
002C  36                PSH A
002D  DE  00G           LDX     R0,D            ; SAVE USP
002F  DF  00G           STX     THGPTR,D
0031  BD  0000G         JSR     AORDEV          ; SET UP ADDRESS FOR THE DEVICE
                                                ; (SETS DEFAULTS)
0034  BD  0000G         JSR     BFRALC          ; ALLOCATE APPROPRIATE BUFFERS.
0037  7F  0000G         CLR     IOFLGS          ; RESET ALL FLAGS
003A  96  00G           LDA A   IOFUNC,D        ; TEST FOR IMAGE IF DOING PRINT
003C  81  0C            CMP A   12.,I
003E  26  11            BNE     PRESTK          ; IF NOT THEN CONTINUE
0040  FE  0000G         LDX     ISB             ; WAS IMAGE DEFINED?
0043  8C  FFFF          CPX     65535.,I
0046  27  09            BEQ     PRESTK
0048  96  00G           LDA A   A.STAT,D        ; SET IMAGE ACTIVE BIT
004A  8A  08            ORA A   #IMACTV,I
004C  97  00G           STA A   A.STAT,D
004E  6D  0000G         JSR     FMTINI          ; INITIALIZE FORMAT IF DEFINED
                        ; STACK INITIALIZATION
                        ; BACK UP STACK UNTIL FIND EOL TAG
                        ; SAVING THE LOCATION OF ALL PERTINENT TAGS
0051  DE  00G   PRESTK: LDX     TAGPTR,D        ; RECALL USP
0053  DF  00G           STX     R0,D
0055  20  03            BRA     AGAIN
                        ;****************** ENTRY POINT FOR BYTE I/O **************
                        ;
                                .GLOBL  SPCIOB
                        ;
0057  86  00G   SPCIOB: LDA A   #ITM2TG,I        ; TAG STACK
0059  36                PSH A
005A  BD  0000G AGAIN:  JSR     BACKUP          ; OK BECAUSE FIRST ONE IS EOLTG
005D  DE  00G           LDX     R0,D            ; LOOK AT TAG
005F  E6  01            LDA B   1,X
0061  C1  00G           CMP B   #TSNTG,I        ; IF FIND  TAG THEN NEED TO STOP
0063  27  08            BEQ     1$
0065  C1  00G           CMP B   EOLTG,I         ; TEST FOR END OF LINE
0067  27  07            BEQ     1$              ; IF END OF LINE GO PROCESS
0069  C1  00G           CMP B   SEMITG,I        ; IF SEMI-TAG THEN JUST PUSH IT
0068  26  06            BNE     TAGIT
006D  37                PSH B
006E  20  EA            BRA     AGAIN
0070  7E  00F4' 1$:     JMP     IOLOPP
0073  9F  00G   TAGIT:  STS     R1,D            ; MAKE IT RELATIVE!
0075  96  00G           LDA A   R0,D            ; GET POINTER (HIGH MEM)
0077  D6  01G           LDA B   R0+1,D
0079  D0  01G           SUB B   R1+1,D          ; SUBTRACT PRESENT (LOW MEM.)
0078  92  00G           SBC A   R1,D
007D  37                PSH B
```

```
 58     007E   36              PSH A
 59     007F   C6    006       LDA B   #CMSIG.#
 60     0081   37              PSH B
 61     0082   20    D6        BRA     AGAIN
```

```
                                .SBTTL  SEARCH LIST FOR VALID ITEM (SET IOFLGS IN PROCESS)
                                ;       THIS SECTION PICKS ITEMS OFF THE STACK AND CHECKS THEM
                                ;
                                        .GLOBL  CRLF
                                        .GLOBL  POINTB,DB       ; POINTERS TO ITEMS FOR PAGE FULL
                                ;
0084    C5      08      GOPROC: BIT B   STGFLG,I        ; TEST IF STRING I/O
0086    26      09              BNE     1$              ; IF SO THEN JSR TO IT
0088    BD      01E7'           JSR     VALUES          ; DO VALUE I/O
008B    96      00G             LDA A   IOFLGS,D        ; SET VALUE FOR PRINT/TAB ALGORITHM
008D    84      DF              AND A   #-1-LASTYP,I
008F    20      07              BRA     END.1
0091    BD      025E'   1$:     JSR     STRNGS          ; DO STRING I/O
0094    96      00G             LDA A   IOFLGS,D        ; SET STRING FOR PRINT/TAB ALGORITHM
0096    8A      20              ORA A   LASTYP,I
0098    97      00G     END.1:  STA A   IOFLGS,D        ; SAVE NEW I/O STATUS
009A    D6      00G             LDA B   PNDEOF,D        ; AN EOF?
009C    26      08              BNE     CLN.2
009E    D6      00G             LDA B   ERRCD,D         ; ANY ERRORS WHILE YOU WERE GONE?
00A0    26      04              BNE     CLN.2           ; YES, SO RETURN
00A2    85      10              BIT A   ENDIO,I         ; END I/O?
00A4    27      4E              BEQ     IOLOPP          ; IF ALL OK GO FOR MORE TAGS
00A6    D6      00G     CLN.2:  LDA B   IOFUNC,D        ; IF PRINT THEN FORCE LAST BUFFER OUT
00A8    C1      0C              CMP B   12.,I           ; IS IT PRINT?
00AA    27      0A              BEQ     3$
00AC    C1      10              CMP B   16.,I           ; SPECIAL I/O FUNCTIONS
00AE    26      1D              BNE     CLN.1
00B0    96      00G             LDA A   A.PRIM,D
00B2    81      21              CMP A   MTPDEV,I
00B4    27      17              BEQ     CLN.1
00B6    96      00G     3$:     LDA A   A.STAT,D        ; IMAGE BEING PROCESSED?
00B8    85      08              BIT A   FMTVLD,I
00BA    27      05              BEQ     2$              ; NOPE
00BC    BD      0000G           JSR     FMTCLN          ; GO CLEAN UP FORMAT
00BF    20      09              BRA     1$              ; GO CLEAN UP BUFFER
00C1    96      00G     2$:     LDA A   IOFLGS,D        ; RETRIEVE SEMIFORS
00C3    85      03              BIT A   SEMIC+MTR,I     ; NO EXTRA "CR" IF MATRIX OR ";" SET
00C5    26      03              BNE     1$
00C7    BD      0000G           JSR     CRLF            ; OUTPUT A "CR"
00CA    BD      0000G   1$:     JSR     SNDBFR          ; FORCE IT
00CD    BD      0000G   CLN.1:  JSR     UNADR           ; UNADDRESS THE DEVICE
0000    9F      00G             STS     R0,D            ; PRUNE STACK TO MY EOL
0002    86      00G             LDA A   ITM2TG,I
0004    BD      0000G           JSR     LOCTG
0007    9E      00G             LDS     R0,D            ; MOVE STACK POINTER TO RIGHT PLACE
0009    32                      PUL A                   ; REMOVE TAG
000A    39                      RTS                     ; RETURN
                                ;*****************************************************************
                                ;       FURTHER PROCESSING
                                ;
0008    D6      00G     IOLOOP: LDA B   IOFLGS,D        ; RETRIEVE FLAGS
000D    C5      01              BIT B   MTR,I           ; IF MATRIX SPECIAL HANDLE
000F    27      A3              BEQ     GOPROC
00E1    30                      TSX
00E2    A6      00              LDA A   0,X             ; OOK AT TAG
00E4    81      00G             CMP A   SEMITG,I        ; IF SEMI TAG THEN SET SEMIC
00E6    26      07              BNE     1$
```

SEARCH LIST FOR VALID ITEM (SET IOFLGS IN PROCESS)

```
                                ORA B   SEMIC,I
                                AND B   #^C<LASTYP,I>   ; CLEAR STRING FLAG
                                STA B   IOFLGS,D
                                INS                     ; GET SEMI TAG OFF STACK
                        1$:     JSR     CRLFEX
                                BRA     GOPROC
                        ;*******************************************************
                        ;       LIST SCANNER
                        IOLOPP: LDA A   ERRCD,D         ; TEST FOR ERRORS
                                BNE     CLN.2           ; GO CLEAN UP
                                LDX     #65535.-1       ; DO SETUP FOR PAGE FULL
                                STX     POINTB
                                STX     DB
                                TSX
                                LDA A   0,X             ; LOOK AT TAG ON STACK
                                LDA B   IOFLGS,D        ; GET FLAGS
                                AND B   #LASTYP,I       ; CLEAR ALL BUT LAST TYPE INFO
                                CMP A   SEMITG,I        ; IF SEMI COLON THEN SET FLAG
                                BNE     1$
                                INS                     ; GET IT OFF THE STACK
                                ORA B   SEMIC,I
                        1$:     STA B   IOFLGS,D        ; SAVE THIS
                                TSX                     ; GET STACK TAG AGAIN
                                LDA A   0,X
                                CMP A   ITM2TG,I        ; DONE?
                                BEQ     CLN.2           ; THEN EXIT
                                INS                     ; REMOVE TAG
                                PUL A
                                PUL B
                                STS     R1,D            ; FIND REAL POSTION
                                ADD B   R1+1,D
                                ADC A   R1,D
                                STA A   R1,D
                                STA B   R1+1,D
                                LDA B   IOFLGS,D
                                LDX     R1,D            ; POINTER TO STACKED ENTRY
                                INX
                                LDA A   0,X             ; LOOK AT THE TAG IN LIST
                                CMP A   VALTG,I         ; VALUE ON STACK
                                BEQ     VALSTK
                                CMP A   #NTINTG,I       ; N.I. NUMERIC
                                BEQ     NTNUM
                                CMP A   #AETG,I         ; ARRAY ELEMENT
                                BEQ     VALARR
                                CMP A   #LOSTG,I        ; LITERAL IN OBJECT STRING
                                BEQ     OBJSTG
                                JMP     NISTRG          ; POINTER TO N.I. STRING
```

```
                                .SBTTL  TAG PROCESSOR
                                ;*****************************************************************
                                ;       VALUE ON STACK
                                ;
013F    08                      VALSTK: INX                     ; IF SO GET POINTER TO VALUE
0140    20  07                          BRA     VALAR1
                                ;*****************************************************************
                                ;       ARRAY ELEMENT
                                ;
0142    EE  03                  VALARR: LDX     3,X             ; ESTABLISH POINTER TO VALUE
0144    4F                              CLR A                   ; TELL SYSTEM THAT IOSNTP IS INVALID
0145    97  00G                         STA A   IOSNTP,D
0147    97  01G                         STA A   IOSNTP+1,D
0149    7E  01E2'               VALAR1: JMP     NXTG.1          ; GET THOSE TAGS!
                                ;*****************************************************************
                                ;       NAME TABLE (NUMERIC)
                                ;
014C    EE  01                  NTNUM:  LDX     1,X             ; RETRIEVE POINTER TO NAME TABLE
014E    DF  00G                         STX     IOSNTP,D        ; SAVE POINTER TO NAME TABLE
0150    A6  04                          LDA A   NTATTR,X        ; GET FLAG BITS
0152    85  40                          BIT A   SCALER,I        ; SCALER?
0154    27  0A                          BEQ     NTARR
0156    85  80                          BIT A   UNDEF,I         ; DEFINED?
0158    27  04                          BEQ     NTVALU
015A    86  40                          LDA A   VALUND,I        ; SET UNDEFINED BIT IN
015C    A7  05                          STA A   NTVAL,X         ; ACTUAL HEADER
015E    20  7F                  NTVALU: BRA     NXTG.3          ; GO FOR NEXT TAG IN LIST
0160    CA  01                  NTARR:  ORA B   MTR,I
0162    D7  00G                         STA B   IOFLGS,D
0164    FF  0000G                       STX     POINTB          ; POINTER TO NAMETABLE FOR ARRAY
0167    EE  07                          LDX     NTWCOL,X        ; COLCNT <- WORKING COL. DIM.
0169    DF  00G                         STX     COLCNT,D
016B    CE  0001                        LDX     1,I             ; INITIALIZE 1 ORIGIN COUNTERS
016E    DF  00G                         STX     LENGTH,D        ; ROWS
0170    DF  00G                         STX     TCOL,D          ; COLS
0172    DF  00G                         STX     TAGPTR,D        ; TOTAL VALUES
0174    DE  00G                         LDX     IOSNTP,D        ; SET UP POINTER TO FIRST VALUE
0176    EE  08                          LDX     NTAPTR,X
0178    20  65                          BRA     NXTG.3          ; GO FOR NEXT TAG IN LIST
                                ;*****************************************************************
                                ;       STRING IN OBJECT LINE
                                ;
017A    EE  03                  OBJSTG: LDX     3,X             ; RETRIEVE POINTER TO STRING
017C    08                              INX
017D    08                              INX
017E    DF  00G                         STX     POINT,D         ; ESTABLISH POINTER TO REAL STRING
0180    09                              DEX
0181    09                              DEX
0182    EE  00                          LDX     0,X             ; GET LENGTH OF STRING
0184    DF  00G                         STX     LENGTH,D
0186    CA  08                          ORA B   STGFLG,I        ; SET STRING FLAG
0188    D7  00G                         STA B   IOFLGS,D
018A    DE  00G                         LDX     CLPTR,D         ; FOR PAGE FULL
018C    FF  0000G                       STX     DB
018F    20  53                          BRA     NXTG.2          ; MORE TAGS?
                                ;*****************************************************************
                                ;       NAME TABLE (STRING)
```

```
                                ;
0191  CA  08      NTSTRG: ORA B   STGFLG,I        ; SET STRING FLAG
0193  D7  00G             STA B   IOFLGS,D
0195  EE  01              LDX     1,X             ; GET POINTER TO NAME TABLE
0197  DF  00G             STX     IOSNTP,D        ; SAVE IT FOR LATER
0199  BD  0000G           JSR     IOPAT1          ; THIS IS A PATCH **********************
                          .GLOBL  IOPAT1
019C  B7  0000G           STA A   DB
019F  A6  04              LDA A   NTATTR,X        ; GET ATTRIBUTE BYTE
01A1  2B  1C              BMI     DEFSTG          ; STRING DEFINED?
01A3  D6  00G             LDA B   A.STAT,D        ; GET I/O OPERATION CODE
01A5  2B  0E              BMI     INSTR           ; IF INPUT - GET MAX. LENGTH
01A7  EE  08              LDX     NTSPTR,X        ; GET POINTER TO STRING HEADER
01A9  A6  02              LDA A   OBJATR,X        ; GET ATRIBUTE BYTE
01AB  85  40              BIT A   64.,I           ; STRING DEFINED?
01AD  26  14              BNE     ERSTRU          ; NOPE
01AF  DE  00G             LDX     IOSNTP,D
01B1  EE  07              LDX     NTWLEN,X        ; IF OUTPUT - GET WORK LENGTH
01B3  20  02              BRA     OUTSTR
01B5  EE  06      INSTR:  LDX     MIDLEN,X        ; GET MAX. LENGTH
01B7  DF  00G     OUTSTR: STX     LENGTH,D        ; STORE IN LENGTH
01B9  DE  00G             LDX     IOSNTP,D        ; RETRIEVE POINTER
01BB  EE  08              LDX     NTSPTR,X        ; GET POINTER TO STRING
01BD  20  20              BRA     NXTG.3          ; GO FOR NEXT TAG IN LIST
01BF  D6  00G     DEFSTG: LDA B   A.STAT,D        ; GET I/O OPERATION CODE
01C1  2B  06              BMI     DIMIT           ; IF INPUT DIMENSION IT 72
01C3  86  00G     ERSTRU: LDA A   ERUNDF,I        ; SET ERROR
01C5  97  00G             STA A   ERRCD,D
01C7  20  13              BRA     CLN.22          ; EXIT
01C9  DE  00G     DIMIT:  LDX     IOSNTP,D        ; RETIEVE NAME TABLE POINTER
01CB  86  48              LDA A   72.,I           ; DEFAULT DIMENSION IS 72
01CD  B7  0001G           STA G   INT1+1
01D0  7F  0000G           CLR     INT1
01D3  BD  0000G           JSR     DIMSTR          ; DIMENSION IT
01D6  DE  00G             LDX     IOSNTP,D        ; GET THE NAME TABLE POINTER AGAIN!
01D8  96  00G             LDA A   ERRCD,D         ; AN ERROR IN DIMENSION?
01DA  27  09              BEQ     INSTR
01DC  7E  0006'   CLN.22: JMP     CLN.2           ; IF ERROR EXIT GRACEFULLY
                          ;**************************************************************
                          ;LOOP BACK TO TEST FOR MORE VALID TAGS BEFORE PROCESSING
                          ;
01DF  BD  0000G   NXTG.3: JSR     ASX
01E2  DF  00G     NXTG.1: STX     POINT,D         ; SAVE THE POINTER TO THE ITEM!
01E4  7E  0DDB'   NXTG.2: JMP     IOLOOP          ; GO FOR MORE TAGS
```

```
                                        .SBTTL  MAIN I/O PROCESSOR (VALUES)
                                ;THIS ROUTINE PREFORMS THE I/O OPERATION DISPATCHING FOR
                                ;THE POINTERS JUST SET UP IN THE I/O PROCESSOR ABOVE.
01E7  86    000G        VALUES: LDA A   RTRNTG,I        ; A TAG!
01E9  36                        PSH A
01EA  96    000G                LDA A   A.STAT,D
01EC  2B    0E                  BMI     3$
01EE  DE    000G                LDX     POINT,D         ; CHECK IF VALUE DEFINED
01F0  A6    00                  LDA A   0,X
01F2  85    40                  BIT A   VALUND,I
01F4  27    06                  BEQ     3$
01F6  86    000G                LDA A   ERUNDF,I
01F8  97    000G                STA A   ERRCD,D
01FA  20    09                  BRA     2$
01FC  CE    0296'       3$:     LDX     VALIO-24.,I     ; SET UP DISPATCH POINTER
01FF  96    000G                LDA A   IOFUNC,D
0201  48                        ASL A                   ; MULT. X 2 FOR OFFSET
0202  BD    0000G               JSR     JMPX            ; JUMP TO VALUE I/O
0205  31                2$:     INS
0206  96    000G                LDA A   A.STAT,D        ; ARE WE DOING INPUT OPERATION
0208  2A    0A                  BPL     4$              ; IF NOT GO FOR MORE TESTS
020A  DE    000G                LDX     IOSNTP,D        ; MAKE SURE VALUE IS MARKED DEFINED
020C  27    06                  BEQ     4$              ; IS IOSNTP VALID?
020E  A6    04                  LDA A   NTATTR,X        ;       EVERYWHERE
0210  84    7F                  AND A   -1-UNDEF,I
0212  A7    04                  STA A   NTATTR,X
0214  8D    6D          4$:     BSR     IOTEST          ; TEST FOR END OR ERRORS
0216  25    45                  BCS     VALEND
0218  96    000G                LDA A   IOFLGS,D        ; CHECK FOR MATRIX I/O
021A  85    01                  BIT A   MTR,I
021C  27    3F                  BEQ     VALEND          ; IF NOT DO NEXT ITEM IN LIST
021E  DE    000G                LDX     POINT,D         ; MOVE TO NEXT VALUE
0220  BD    0000G               JSR     A8X
0223  DF    000G                STX     POINT,D
                                ; ARRAY OUTPUT ---- ALL COUNTERS ARE 1 ORIGIN
                                ; TCOL   = PRESENT COL. NUMBER
                                ; LENGTH = PRESENT ROW NUMBER
                                ; COLCNT = WORKING COL. COUNT ; IF=1 THEN VECTOR
                                ; TAGPTR = TOTAL # OF VALUES PRINTED
                                ; NTWROW,IOSNTP = PRESENT WORKING LENGTH
                                ;
0225  DE    000G                LDX     COLCNT,D        ; SEE IF VECTOR
]]]]]
0227  27    0B                  BEQ     VCTCHK          ; GO SEE IF DONE
0229  DE    000G                LDX     TCOL,D          ; SEE IF DONE WITH THIS ROW
022B  9C    000G                CPX     COLCNT,D
022D  27    12                  BEQ     NEWROW
022F  08                        INX
0230  DF    000G                STX     TCOL,D
0232  20    22                  BRA     NXTVAL          ; GO DO NEXT VALUE
0234  DE    000G        VCTCHK: LDX     IOSNTP,D        ; TEST IF DONE ALL THE VECTOR
0236  EE    05                  LDX     NTWROW,X
0238  9C    000G                CPX     TAGPTR,D
023A  26    1A                  BNE     NXTVAL          ; GO DO NEXT VALUE
023C  BD    029A'               JSR     CRLFFX          ; PUT IN A BLANK LINE
023F  20    1C                  BRA     VALEND
                                ;
0241  BD    029A'       NEWROW: JSR     CRLFFX          ; PUT IN A BLANK LINE
```

```
    58  0244  CE  0001          LDX   1,I          ; INITIALIZE COL. COUNTER
    59  0247  DF  00G           STX   TCOL,D
    60  0249  DE  00G           LDX   IOSNTP,D     ; CHECK IF DONE
    61  024B  EE  05            LDX   NTWROW,X
    62  024D  9C  00G           CPX   LENGTH,D
    63  024F  27  0C            BEQ   VALEND       ; DONE
    64  0251  DE  00G           LDX   LENGTH,D     ; ELSE INC. ROW NUMBER
    65  0253  08                INX
    66  0254  DF  00G           STX   LENGTH,D
    67                        ;
    68  0256  DE  00G   NXTVAL: LDX   TAGPTR,D     ; UPDATE TOTAL COUNT
    69  0258  08                INX
    70  0259  DF  00G           STX   TAGPTR,D
    71  025B  20  8A            BRA   VALUES
    72  025D  39        VALEND: RTS
```

```
                                        .SBTTL  MAIN I/O PROCESSOR (STRINGS)
                                ;THIS ROUTINE ALSO HANDLES THE CONTROL FOR I/O GIVEN THE POINTERS ETC
                                ;THAT WERE DEFINED ABOVE
                                ;
025E  86  00G     STRNGS: LDA A   RTRNTG,I            ; TAG IT
0260  36                  PSH A
0261  CE  02A4'            LDX     STGIO-24.,I       ; SET UP DISPATCH POINTER
0264  96  00G             LDA A   IOFUNC,D
0266  48                  ASL A                       ; SET PROPER OFFSET
0267  BD  0000G           JSR     JMPX                ; JUMP TO STRING I/O
0269  31                  INS
026A  96  00G             LDA A   A.STAT,D
026D  2A  2A              BPL     STREXX              ; INPUT LIKE OPERATION?
026F  DE  00G     UPDLEN: LDX     IOSNTP,D            ; GET POINTER TO NAME TABLE

                                ;       ***** START OF CHANGE BY GEORGE

0271  07                  TPA
0272                      SEI
0272  01  0F              BYTE    01,17
0274  D6  00G             LDA B   CHRCNT,D            ; SET IN NEW WORKING LENGTH
0276  E7  07              STA B   NTWLEN,X
0278  D6  01G             LDA B   CHRCNT+1,D
027A  E7  08              STA B   NTWLEN+1,X
027C  EE  08              LDX     NTSPTR,X            ; UPDATE ATTR. IN STRING HEADER
027E  C6  20              LDA B    HO2D,I ; STRING
0280  E7  02              STA B   OBJATR,X
0282  06                  TAP                         ;TURN THEM BACK ON
                                ; **** END OF CHANGE
0283  0C                 IOTEST: CLC                  ; ASSUME NO ERRORS
0284  96  00G             LDA A   ERRCD,D             ; TEST FOR ERRORS
0286  9A  00G             ORA A   PNDEOF,D            ; OR END OF FILES
0288  26  08              BNE     BYEBYE
028A  96  00G             LDA A   CRSTAT,D            ; GET CR STATUS
028C  2A  08              BPL     STREXX              ; EXIT IF NOT VALID
028E  85  38              BIT A   EOFTYP,I            ; TEST FOR END OF FILES
0290  27  07              BEQ     STREXX              ; OK
0292  96  00G     BYEBYE: LDA A   IOFLGS,D            ; SET TO END I/O
0294  8A  10              ORA A   ENDIO,I
0296  97  00G             STA A   IOFLGS,D
0298  0D                  SEC                         ; TELL PEOPLE IT'S AN ERROR EXIT
0299  39                 STREXX: RTS
```

```
                                        .SBTTL  CRLFFX--CRLFLF FEATURE TESTS IF PRINT
                                ;
029A  96  00G     CRLFFX: LDA A   IOFUNC,D         ; A PRINT?
029C  81  0C              CMP A   12.,.
029E  26  00              BNE     1$
02A0  96  00G             LDA A   A.STAT,D
02A2  85  08              BIT A   FMTVLD,I
02A4  26  07              BNE     1$
02A6  86  00G             LDA A   RTRNTG,I
02A8  36                  PSH A
02A9  BD  0000G           JSR     CRLFLF
02AC  31                  INS
02AD  39          1$:     RTS
```

```
                                        .SBTTL  I/O DISPATCH TABLES
                                ; MAY REORDER THIS TABLE
                                        .GLOBL  WBYTVL,RBYTVL
                                ;
027E    0000G           VALIO:  .WORD   PRIVAL          ; VALUE I/O DISPATCH TABLE
0280    0000G                   .WORD   INPVAL
0282    0000G                   .WORD   REAVAL
0284    0000G                   .WORD   WRIVAL
0286    0000G                   .WORD   PRIVAL          ; SPECIAL FUNCTION DISPATCH
0288    0000G                   .WORD   WBYTVL
028A    0000G                   .WORD   RBYTVL
                                ;
028C    0000G           STGIO:  .WORD   PRISTG          ; STRING I/O DISPATCH TABLE
028E    0000G                   .WORD   INPSTG
0290    0000G                   .WORD   REASTG
0292    0000G                   .WORD   WRISTG
0294    0000G                   .WORD   PRISTG          ; SPECIAL FUNCTION DISPATCH
        0001'                   .END
```

SYMBOL TABLE

```
ABRFLG=  0040        ADRDEV= XXXXXX G     AFAIL =  0030        AGAIN   005AR       AMAT  =  0010
ARRAY =  0020        ASTR  =  0026        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     ASX   = XXXXXX G     ASX   = XXXXXX G     BACKUP= XXXXXX G
BAKSTG= XXXXXX G     BANK  = XXXXXX G     BERRLG= XXXXXX G     BERSTT=  0020        BLINK = XXXXXX G
BMAT  =  0004        BRKCNT= XXXXXX G     BSTR  =  0008        BUSACT=  0010        BYEBYE   0292R
CDOPTR= XXXXXX G     CDSPTR= XXXXXX G     CHAR  = XXXXXX G     CHRCNT= XXXXXX G     CLN.1    00CDR
CLN.2    0096R       CLN.22   01DCR       CLPTR = XXXXXX G     CMAT  =  0001        COLCNT= XXXXXX G
CRDC3 =  0002        CREOF =  0008        CREOI =  0004        CREOT =  002C        CRETX =  0010
CRLF  = XXXXXX G     CRLFFX   0298R       CRLFLF= XXXXXX G     CRNORM=  0001        CRSTAT= XXXXXX G
CRWLD =  0080        CSTR  =  0002        CTKN  = XXXXXX G     CURSOR= XXXXXX G     DATDEV=  0022
DB    = XXXXXX G     DEFSTG   018FR       DIMFLG=  0004        DIMIT    01C9R       DIMSTR= XXXXXX G
DIRCT =  0080        DISCNT= XXXXXX G     DISSRQ=  0008        DL    = XXXXXX G     DP    = XXXXXX G
DREXTA= XXXXXX G     DREXTB= XXXXXX G     DSPDEV=  0020        DT    = XXXXXX G     EDTBFR= XXXXXX G
ENDIO =  0010        ENDKEY=  0040        END.1    0098R       EOFTYP=  0038        EOLTG = XXXXXX G
EOSTG = XXXXXX G     ERATSN= XXXXXX G     ERDOMN= XXXXXX G     EREOM = XXXXXX G     ERFBFR= XXXXXX G
ERFILE= XXXXXX G     ERIOE = XXXXXX G     ERNIOD= XXXXXX G     ERNSEP= XXXXXX G     ERRCD = XXXXXX G
ERSTRU   01C3R       ERTERM= XXXXXX G     ERUNDF= XXXXXX G     ESTG  = XXXXXX G     EXTFLG=  0080
FILDEV=  0000        FIX1H =  0003        FIX1B =  0004        FMTCLN= XXXXXX G     FMTINI= XXXXXX G
FMTWLD=  0008        FNFLG =  0010        FORTG = XXXXXX G     GLBFLG= XXXXXX G     GOPROC   0084R
GOSTG = XXXXXX G     IMXTG = XXXXXX G     INPSTG= XXXXXX G     INPUTK=  0001        INPVAL= XXXXXX C
INSTR    0185R       INT1  = XXXXXX G     IOBFR1= XXXXXX G     IOCLNR   0008RG      IOFLGS= XXXXXX G
IOFUNC= XXXXXX G     IOLOOP   0006R       IOLOPP   00E4R       IOPRI1= XXXXXX G     IOPROC   0000RG
IOSCAN   002ARG      IOSNTP= XXXXXX G     IOTEST   0283R       ISB   = 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     LDRX  = XXXXXX G     LDXX  = XXXXXX G     LENGTH= XXXXXX G
LISTTG= XXXXXX G     LNNOTG= XXXXXX G     LOCTG = XXXXXX G     LSP   = XXXXXX G     LSTFMT=  0002
MTBER = XXXXXX G     MTPDEV=  0021        MTPD2 =  0023        MTR   =  0001        NEWROW   0241R
NLPTR = XXXXXX G     NOKEY =  0080        NOOUT = XXXXXX G     NOWRIT=  0001        NTAPTR=  0008
NTARR    0160R       NTATTR=  0004        NTDIMS=  0009        NTDLEN=  0005        NTLINK=  0000
NTNAME=  0002        NTNUM    014CR       NTPTR = XXXXXX G     NTRELV=  0010        NTSPTR=  0008
NTSTRG   0191R       NTVAL =  0005        NTVALU   015ER       NTWCOL=  0007        NTWLEN=  0007
NTWROW=  0005        NULLTG= XXXXXX G     NXTG.1   01E2R       NXTG.2   01E4R       NXTG.3   01DFR
NXTVAL   0256R       OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000
OBJSTG   017AR       ONSFLG=  0002        OPRADR= XXXXXX G     OUTSTR   0187R       PRETG = XXXXXX G
PARM  =  0008        PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMFP =  0003
PGMLEN=  0000        PGMNM=   0007        PGMPTR= XXXXXX G     PGMTG = XXXXXX G     PLOSTG= XXXXXX G
PNDEOF= XXXXXX G     PNDFLG= XXXXXX G     PNTNTG= XXXXXX G     PNTSTG= XXXXXX G     POINT = XXXXXX G
POINTB= XXXXXX G     PPMODE= XXXXXX G     PRESTK   0051R       PRIDEF=  0001        PRISTG= XXXXXX G
PRIVAL= XXXXXX G     PRTTG = XXXXXX G     PSCTG = XXXXXX G     PUTBYT= XXXXXX G     RBYTE = XXXXXX G
RBYTVL= XXXXXX G     REASTG= XXXXXX G     REAVAL= XXXXXX G     RECLFG=  0004        RFAIL =  00C0
RMAT  =  0040        RPTCTL=  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        SECDEF=  0002        SEMIC =  0002        SEMITG= XXXXXX G
SNOBFR= XXXXXX G     SNOIT =  0080        SPCIOB   0057RG      STRT37= XXXXXX G     STGFLG=  0008
STGIO    02BCR       STPFLG=  0040        STPKEY=  0020        STREXX   0299R       STRING=  0010
STRNGS   025ER       SYSERR= XXXXXX G     TAGIT    0073R       TAGPTR= XXXXXX G     TCOL  = XXXXXX G
TRCFLG=  0020        TS1DEV=  0025        UNADR = XXXXXX G     UNDEF =  0080        UPDLEN   026FR
VALARR   0142R       VALAR1   0149R       VALEND   0250R       VALIO    02AER       VALSTK   013FR
VALTG = XXXXXX G     VALUES   01E7R       VALUND=  0040        VCTCHK   0234R       WBYTE = XXXXXX G
WBYTVL= XXXXXX G     WRISTG= XXXXXX G     WRIVAL= XXXXXX G     XAXIS = XXXXXX G
. ABS.  0000    00
```

```
IOPROC  I/O PROCESSOR  RT-11 MMAC  VM02-10   14-OCT-76 01:35:15 PAGE 10+
SYMBOL TABLE

        02C6      01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2314. WORDS
,SY:IOPROC/C<DK1:SE1CL1,IOPROC
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

```
DEFSTG  6-67    6-82#
DIMFLG  1-28#
DIMIT   6-83    6-87#
DIMSTR  1-298#  6-91
DIRCT   1-200#
DISCHT  1-221#
DISSLO  1-36#
DL      1-233#
DP      1-232#
DREXTA  1-271#  1-291#  3-16*  3-18*  3-23
DREXTB  1-272#
DSPDEV  1-256#
DT      1-234#
EDTBFR  1-265#
END.1   5-12    5-16#
ENDLO   1-219#  5-21    8-33
ENDKEY  1-213#
EOFTYP  1-186#  8-35
EOLTG   1-142#  2-4     3-11    3-20    4-45
EOSTG   1-143#
ERATSN  1-170#
ERDOMN  1-164#
EREOM   1-171#
ERFBFR  1-167#
ERFILE  1-169#
ERIOE   1-168#
ERNIOD  1-173#
ERNSEP  1-165#
ERRCD   1-42#   5-19    5-66    6-85*  6-93    7-13*  8-30
ERSTRU  6-73    6-84#
ERTERM  1-166#
ERUNDF  1-172#  6-84    7-12
ESTG    1-124#
EXTFLG  1-25#
FILDEV  1-254#
FIX1A   1-117#
FIX1B   1-118#
FMTCLN  1-305#  5-34
FMTINI  1-304#  4-27
FMTVLD  1-203#  4-25    5-32    9-7
FNFLG   1-27#
FORTG   1-126#
G.BFLG  1-31#
GOPROC  5-7#    5-53    5-63
GOSTG   1-125#
IMXTG   1-129#
INFSTG  1-290#  10-14
INPUTK  1-219#
INPVAL  1-289#  10-6
INSTR   6-69    6-77#   6-94
INT1    1-297#  6-89*   6-90*
IOBFR1  1-266#
IOCLNR  1-278#  2-8     3-8#
IOFLGS  1-246#  4-17*   5-10    5-14    5-16*  5-36    5-51    5-60*  5-73    5-79*  5-92    6-29*  6-52*  6-60*
        2-29    8-37    8-39*
IOFUNC  1-190#  4-18    5-23    7-16    8-8     9-3
IOLOOP  5-51#   6-101
```

CROSS REFERENCE TABLE (CREF V01-03)

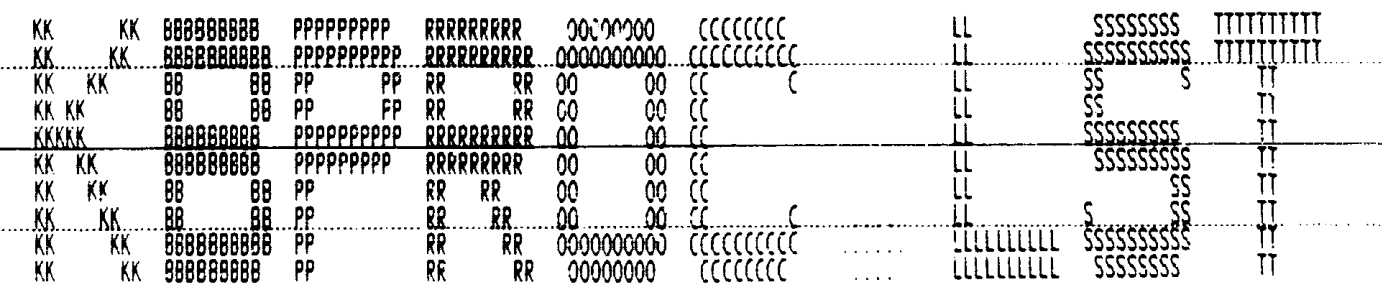
CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
R8      1-38#
R9      1-38#
RBYTE   1-289#
RBYTVL  10-3#   10-11
REASTG  1-290#  10-15
REAVAL  1-289#  10-7
RECLFG  1-217#
RFAIL   1-152#
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNTG  1-140#  3-8     7-4     8-5     9-9
RUNFLG  1-32#
RUNN    1-218#
SBP     1-47#
SCALER  1-82#   6-21
SECDEF  1-205#
SEMIC   1-317#  5-37    5-58    5-78
SEMITG  1-144#  4-47    5-56    5-75
SNOBFR  1-301#  5-40
SNDIT   1-228#
SPC108  4-36#   4-38#
STAT37  1-247#
STGFLG  1-318#  5-7     6-51    6-59
STGIO   8-7     10-13#
STPFLG  1-33#
STPKEY  1-214#
STREXX  8-13    8-34    8-36    8-41#
STRING  1-84#
STRNGS  5-13    8-5#
SYSERR  1-43#
TSTDEV  1-261#
TAGIT   4-48    4-52#
TAGPTR  1-279#  4-13#   4-31    6-36#   7-52    7-68    7-70#
TCOL    1-245#  6-35#   7-44    7-48#   7-59#
TRCFLG  1-34#
UNADR   1-303#  5-41
UNDEF   1-81#   6-23    7-25
UPDLEN  8-14#
VALAR1  6-6     6-14#
VALARR  5-101   6-10#
VALEND  7-28    7-31    7-55    7-63    7-72#
VALIO   7-15    10-5#
VALSTK  5-97    6-5#
VALTG   1-134#  5-96
VALUES  5-9     7-4#    7-71
VALUND  1-88#   6-25    7-10
VCTCHK  7-43    7-50#
WBYTE   1-289#
WBYTVL  10-3#   10-10
WRISTG  1-290#  10-16
WRIVAL  1-289#  10-8
XAXTS   1-248#
```

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-35 8-19



14-OCT-76

TABLE OF CONTENTS

```
                        .TITLE  KBPROC KEYBOARD PROCESSOR
                        .IDENT  /SBA051/
                        ;
                        ;
                        ;PURPOSE---- PROCESS A KEY PRESSED ON THE KEYBOARD AND DISPATCH
                        ;            ACCORDING TO THE STATUS OF THE SYSTEM
                        ;
                        ;ENTER WITH- RUNN   -  KEYBOARD STATUS  0 = WAITING FOR COMMAND LINE
                        ;                                       1 = RUNNING PROGRAM
                        ;            INPUTK - DOING INPUT FROM KEYBOARD
                        ;                     0 = NOT DOING INPUT LINE
                        ;                     1 = PROGRAM RUNNING, WAITING FOR
                        ;                       KEYBOARD INPUT
                        ;            KEYSTK - NUMBER OF ENTRIES PRESENTLY ON THE USER DEFINE
                        ;                     KEY STACK (0 THRU 7 OCTAL)
                        ;            CC     = CHARACTER COUNT FROM EDITOR
                        ;
                        ;EXIT WITH-- KEYSTK  -  POTENTIALLY MODIFIED
                        ;            ENDKEY  -  SET, IF AN END-OF-LINE IS ENTERED:
                        ;                       RETURN
                        ;                       BREAK
                        ;                       USER DEFINE (UNDER INPUT)
                        ;            KEYRUN  -  KEYBOARD DECODER'S RUN FLAG
                        ;                       0 = ALL KEYS ACTIVE
                        ;                       1 = IMMEDIATE EXECUTE ONLY
                        ;                      (KEYRUN = RUNN OR (NOT INPUTK) )
                        ;
                        ;FLAGS------ BREAK - SET IF ALLREADY HAD ONE BREAK
                        ;            NOKEY - USER DEFINE KEYS DISABLED
                        ;            NTRELY - THE "NOT REALLY FLAG", SET IF CAN'T BE INTERRUPTED
                        ;
                        ;
                        ;; GLOBAL SUBROUTINES
                                .GLOBL  TYPIN           ; THE EDITOR
                                GLOBL   TSTINT          ; EVALUATOR TEST INTERUPTS
                                .GLOBL  LNEVL           ; EVALUATOR EXECUTE A SINGLE LINE
                                .GLOBL  TRANSL          ; THE TRANSLATOR
                                GLOBL   END             ; EVALUATOR TO END TH PROGRAM
                                .GLOBL  PRTERR          ; THE ERROR HANDLER
                                .GLOBL  PCHAR           ; DISPLAY SINGLE CHARACTER
                                GLOBL   DSPCPY          ; MAKE HARD COPY OF DISPLAY
                                .GLOBL  DSPSTT          ; DSIPLAY DRIVER STATUS
                                .GLOBL  GETKEY          ; EXTRACTS KEY CODE FROM THE HARDWARE
                                GLOBL   PGCTR           ; PAGE SCREEN COUNTER
                                .GLOBL  KBQIN           ; ENTER KEY INTO THE QUE
                                .GLOBL  KBQOUT          ; EXTRACT KEY FROM THE QUE
                                GLOBL   KBINT           ; KEY BORAD INTERUPT PROCESSOR
                                .GLOBL  KBPROC          ; PROCESS A SINGLE KEY
                                .GLOBL  IDLE            ; THE MASTER IDLE LOOP
                                GLOBL   CIDLE           ; THE CURSOR IDLE LOOP
                                .GLOBL  GENCUR          ; TO OUTPUT THE CURSOR
                                .GLOBL  ERRCOB          ; BACKUP ERROR CODE
                                GLOBL   INPUT           ; INPUT CODE
                                .GLOBL  ADRDEV          ; ADDRESS DEVICE
                                .GLOBL  BFRALC          ; FIND BUFFER SPACE
                                GLOBL   OPRADR          ; CONSTANT FOR I/O CONTROLLER
                        ;
```

```
                                ;       SOME CONSTANTS AND SUCH
                                ;
                                .GLOBL  KEYQUE          ; THE TYPE AHEAD QUE
                                .GLOBL  QEND            ; END OF QUE
                                .GLOBL  QIN             ; INPUT POINTER FOR THE QUE
                                .GLOBL  QOUT            ; OUTPUT POINTER FOR THE QUE
                                .GLOBL  PIALT           ; THE LIGHTS PIA REGISTER
0080                    BSYLT   = H80                   ; THE BUSY LIGHT
0040                    IOPLT   = H40                   ; THE I/O LIGHT
0020                    BRKLT   = H20                   ; THE BREAK LIGHT
                                ;
                                ;KEYBOARD KEYCODES
0080                    NOKEY   = H80                   ;INVALID KEY
00EA                    PAGEKY  = H0EA                  ;PAGE
00CA                    RSTKY   = H0CA                  ;RESET
00ED                    MKCPYK  = H0ED                  ;MAKE COPY
00EC                    RWNDKY  = H0EC                  ;REWIND
00F6                    STEPKY  = H0F6                  ;STEP PROGRAM
00EB                    BRKKY   = H0EB                  ;BREAK
00E9                    MAXUKY  = H0E9                  ;MAXIMUM NUMBER ON USER KEY
00C0                    MINUKY  = H0C0                  ;MINIMUM NUMBER ON USER KEY CODE
0020                    SHIFTK  = H020                  ;SHIFT POSITION FOR USER KEYS
                                ;
```

```
                                        .SBTTL  KBINT KEYBOARD INTERUPT HANDLER
                                        .GLOBL  INITMT
                                        .GLOBL  IECOFF,IECIFC,UNADR
                                        .GLOBL  MSKCTR
                                        .GLOBL  EDTMAX
                                        .GLOBL  EDTPTR
                                        .GLOBL  CRTRST
                                        .GLOBL  MTAFIN
                                        .GLOBL  MTPRWD          ;MAG TAPE REWIND SUBROUTINE
                                ;
0000  80  0000G       KBINT:    JSR     GETKEY          ; GET THE KEYCODE
0003  96  00G                   LDA A   KBIN,D
0005  81  80                    CMP A   NOKEY,I         ; IS IT A VALID KEY?
0007  26  01                    BNE     1$
0009  39                        RTS                     ; JUST RETURN IF NOT
000A  40              1$:       TST A                   ; OTHERWISE PROCESS IT
000B  2B  03                    BMI     IMMEX           ; IF IMMEDIATE EXECUTE KEY WORK AWHILE
000D  7E  01BC'                 JMP     KBQIN           ; ELSE JUST Q IT
                                ;*******************************************************************
                                ;       IMMEDIATE EXECUTE KEY PROCESSOR
0010  81  EA          IMMEX:   CMP A   PAGEKY,I        ; PAGE KEY
0012  26  04                    BNE     RST
0014  86  0C                    LDA A   14,I            ; "FF" FOR DISPPLAY
0016  20  06                    BRA     RST.1
0018  81  CA          RST:     CMP A   RSTKY,I         ; RESET KEY
001A  26  05                    BNE     CPY
001C  86  1E                    LDA A   36,I            ; "RS" FOR DISPLAY
001E  7E  0000G       RST.1:    JMP     PCHAR
0021  81  ED          CPY:     CMP A   MKCPYK,I        ; MAKE COPY KEY
0023  26  18                    BNE     USERKY
0025  7E  0000G                 JMP     DSPCPY          ; MAKE HARD COPY
0028  B6  0000G       MTPRWD   LDA A   PIALT           ; TURN ON I/O LIGHT
002B  36                        PSH A
002C  8A  40                    ORA A   IOPLT,I
002E  B7  0000G                 STA A   PIALT
0031  BD  0000G                 JSR     INITMT          ; INITIALIZE MAGTAPE
0034  5F                        CLR B
0035  BD  0000G                 JSR     MTAFIN
0038  7F  0000G                 CLR     ERRCD           ; IF ERROR IGNORE IT
003B  32                        PUL A
003C  B7  0000G                 STA A   PIALT
003F  39                        RTS
                                ;*******************************************************************
                                ;       USER DEFINED KEYS
                                ;A USER DEFINED KEY WILL ALWAYS BE PLACED ON THE KEYSTK IF
                                ;       THERE IS ROOM.  IT ALSO PUTS A CODE (UKYFLG)
                                ;       INTO THE QUE FOR USE LATER.  IT IS UP TO SOME OTHER ROUTINE TO
                                ;       DECIDE WHAT AND WHEN ACTION IS TO BE TAKEN ON THIS KEY.
                                ;
0040  81  EA          USERKY:  CMP A   MAXUKY+1,I      ; USER DEF KEY
0042  22  1D                    BHI     SPCKEY          ; IF NO THEN TRY OTHERS
0044  81  C0                    CMP A   MINUKY,I
0046  25  19                    BCS     SPCKEY
0048  80  BF          STKIT:   SUB A   MINUKY-1,I      ; EXTRACT KEYNUMBER 0-19
004A  16                        TAB
004B  84  0F                    AND A   #0F,I ; CLEAR HIGH HALF WORD
004D  C5  20                    BIT B   SHIFTK,I        ; TEST FOR SHIFT KEY
```

```
                                        BNE     STACKK
                                        ADD A   10,I            ; ADD 10  IF SHIFT KEY
0053  CE  FFFFG   STACKK: LDX     KEYSTK-1,I
0056  08          1$:     INX
0057  60  00              TST     0,X
0059  26  FB              BNE     1$
005B  A7  00              STA A   0,X
005D  7F  0008G           CLR     KEYSTK+8        ; MAKE SURE LAST BYTE IS ZERO
0060  39                  RTS
0061  81  EB      SPCKEY: CMP A   BRKKY,I         ; IS IT THE BREAK KEY?
0063  27  03              BEQ     BRKKEY
0065  7E  01BC'           JMP     KBQIN           ; QUE IT IF NOT
                        ;**************************************************************
                        ;       BREAK KEY PROCESSOR
                        ;HERE IT WILL ALWAYS JUST QUE UP THE BREAK KEY AND SET THE
                        ;       EVALUATOR INTERUPT BIT.
                        ;LATER IT WILL TAKE THE FOLLOWING ATION AS IT COMES OUT OF THE
                        ;       QUE:
                        ;IF NOT IN RUN MODE THE BREAK KEY IS COMPLETELY IGNORED.
                        ;IF RUNNING
                        ;       THE FIRST BREAK WILL BE QUED AND WILL SET THE EVALUATOR
                        ;       ABORT BIT.
                        ;       THE SECOND TIME THE QUE WILL BE ZEROED AND THE PROGRAM
                        ;       ABORTED (IF NTRELY-THE NOT REALLY FLAG-ISN'T SET).
                        ;
0068  DE  00G     BRKKEY: LDX     QOUT,D          ; CLEAN UP QUE
006A  DF  00G             STX     QIN,D
006C  D6  00G             LDA B   KBFLAG,D        ; TEST IF IN RUNN MODE
006E  C5  02              BIT B   RUNN,I
0070  27  10              BEQ     BRK1
0072  D6  00G             LDA B   PNDFLG,D        ; CHECK ON BREAK FLAG
0074  2B  00              BMI     BRKSET
0076  CA  80              ORA B    ABO,I  ; SET IT
0078  D7  00G             STA B   PNDFLG,D
007A  F6  0000G           LDA B   PIALT           ; TURN ON BREAK LIGHT
007D  CA  20              ORA B   BRKLT,I
007F  F7  0000G           STA B   PIALT
0082  39          BRK1:   RTS
0083  7D  0000G   BRKSET: TST     MSKCTR          ; TEST IF OK TO TERMINATE
0086  26  FA              BNE     BRK1
0088  4F                  CLR A
0089  97  00G             STA A   A.STAT,D
0088  97  00G             STA A   PNDFLG,D
008D  BD  0000G           JSR     IECIFC
0090  86  F0              LDA A    AOF0,I         ;CLEAR LINE LENGTH
0092  B7  0000G           STA A   LINELN
                          .GLOBL  LINELN
0095  DE  00G             LDX     ZX,D            ; SET EDIT BUFFER CLEAR
0097  DF  00G             STX     EDTPTR,D
0099  DF  00G             STX     EDTMAX,D
                          .GLOBL  ZX
0098  86  00G             LDA A   ERTERM,I        ; SET UP TERMINATION ERROR
0090  36                  PSH A                   ; FOR UNDO!!!!!
                        ;>>>>>> MODIFY RTI STACK ENTRY TO DO A JUMP TO END!!!!!
009E  30                  TSX
009F  6F  03              CLR     3,X             ; CLEAR BANK
00A1  B6  0080'           LDA A   ENDIT
```

.KBINT KEYBOARD INTERUPT HANDLER

```
115  00A4  A7  09          STA A  9.,X
116  00A6  B6  00B1'       LDA A  ENDIT+1
117  00A9  A7  0A          STA A  10.,X
118  00AB  96  00G         LDA A  A.PRIM,D      ; SET UP ACC A FOR UNADR
119  00AD  7E  0000G       JMP    UNADR3        ; GO UNADRESS !!!!! XXXXXXPATCHXXXXXXXXXXXXX
120                        .GLOBL UNADR3
121  00B0  0000G     ENDIT: .WORD  END
```

```
                                        .SBTTL  KBPROC SINGLE KEY PROCESSOR
                                ;THIS ROUTINE PROCESSES KEYCODES AFTER THEY HAVE BEEN THRU
                                ;       THE INITIAL KEYBOARD INTERUPT PROCESSOR.  WHEN CALLED
                                ;       THE CHARACTER IS IN A.
                                ;
0000                            CR      = 000           ; "CR"
                                ;
0082  40                        KBPROC: TST A                   ; RESET CONDITION CODES
0083  28  06                     BMI     IMMED           ; IMMEDIATE TYPE KEY?
0085  7F  0000G                 EDITIT: CLR     BLINK
0088  7E  0000G                 JMP     TYPIN           ; ELSE EDIT IT
                                ;*********************************************************
                                ;       IMMEDIATE EXECUTE KEYS
                                ;1) STEP SETS THE STEP KEY BIT IN KBFLAG FOR USE LATER
                                ;2) AUTO LOAD SETS THE AUTO LOAD BIT IN KBFLAG
                                ;ALL OTHERS ARE PASSED TO EDIT (AND IT BETTER KNOW WHAT TO DO)
                                ;
0088  81  EC                    IMMED:  CMP A   RWNDKY,I        ; REWIND KEY
008D  26  06                            BNE     OTHKEY
008F  BD  0028'                         JSR     MTPRWD
00C2  7E  0000G                 JMP     CRTRST
00C5  81  F6                    OTHKEY: CMP A   STEPKY,I        ; IS IT A STEP KEY?
00C7  26  EC                            BNE     EDITIT          ; IF NOT THEN EDIT KEY
00C9  D6  000G                  LDA B   KBFLAG,D        ; GET KEYBOARD STATUS
00CB  C5  01                    BIT B   INPUTK,I
00CD  26  04                            BNE     1$              ; BRANCH IF IN INPUT MODE
00CF  CA  20                    ORA B   STPKEY,I
0001  D7  000G                  STA B   KBFLAG,D        ; SET STEP FLAG
0003  39                        1$:     RTS
```

```
                                        .SBTTL  IDLE MAIN MONITOR IDLE LOOP
                                ; ENTRY TO THS ROUTINE DOES NECESSORY
                                ; INITIALIZATION FOR EVALUATOR END.  CHECKS FOR
                                ; ANOY ERRORS, AND ANY LATENT ON CONDITIONS.
                                ;
                                        .GLOBL  AUTONO          ; AUTO NUMBER FOR TYPIN
                                        .GLOBL  EDTCLS
                                        .GLOBL  QTRNSL          ; WAIT TILL YOU FIND OUT WHAT IT DOES
                                        .GLOBL  PIASRQ
                                        .GLOBL  KYRATE
                                        .GLOBL  XEQSTK,XEQSP    ; THESE ARE MEANINGFUL?????????
                                ;
0004    86      03      IDLE:   LDA A   3.,I            ; SET MAXIMUM ACCELERATION FOR KEYBOARD
0006    B7      0000G           STA A   KYRATE
0009    CE      0002G           LDX     XEQSTK+2,I
000C    DF      00G             STX     XEQSP,D
000E    7F      0000G           CLR     BLINK           ; MAKE SURE THE SYSTEM IS RESET
0011    BD      0000G           JSR     TSTINT
0014    7F      0000G           CLR     KBFLAG          ; SO PAGE FULL WON'T BE CALLED
0017    BD      0000G           JSR     PRTERR          ; ERROR PROCESSOR
001A    4F                      CLR A                   ; CLEAR SOME STATUS
001B    97      00G             STA A   KBFLAG,D
001D    97      00G             STA A   LCLFLG,D
001F    97      00G             STA A   A.STAT,D
0021    96      00G             LDA A   GLBFLG,D
0023    84      3F              AND A   -1-(RUNFLG!STPFLG),I    ; CLEAR EVAL'S STATUS
0025    97      00G             STA A   GLBFLG,D
                                        .GLOBL  IDLVCT          ; IDLE LOOP INTERUPT VECTOR
0027    B6      0002G           LDA A   IDLVCT+2
002A    27      00              BEQ     1$
002C    BD      0000G           JSR     SETBNK
                                        .GLOBL  SETBNK
002F    FE      0000G           LDX     IDLVCT
0032    AD      00              JSR     0,X
0034    BD      0000G           JSR     CLRBNK
                                        .GLOBL  CLRBNK
0037    20      CB              BRA     IDLE
0039    CE      0000G   1$:     LDX     INPUT,I
003C    DF      00G             STX     OPRADR,D
003E    BD      0000G           JSR     ADRDEV
0041    7F      0000G           CLR     PIALT           ; TURN OFF I/O LIGHT
0044    86      98              LDA A   #98,I   ; SET ALL LINES HIGH ON THE BUS
0046    B7      0000G           STA A   PIASRQ
0049    BD      0000G           JSR     BFRALC
004C    DE      00G             LDX     A.MAX,D         ; PATCH FOR LINE RECALL ****
004E    09                      DEX                     ; SO UNCOMPILE WILL GO TO COMPRESSED FORM.
004F    DF      00G             STX     A.MAX,D
0051    01                      NOP                     ; SO IT WILL ALL FIT
0052    86      80              LDA A   #80,I   ; GET THE CURSOR
0054    97      00G             STA A   CURSOR,D
0056    96      00G             LDA A   PNDFLG,D        ; TURN OFF BREAK
0058    84      7F              AND A   177,I
005A    97      00G             STA A   PNDFLG,D        ; IF PENDING INTERRUPT
005C    BD      0000G           JSR     AUTONO          ; DOES AUTO NUMBER
005F    BD      017E'           JSR     CIDLE           ; GO GET A LINE FROM KEYBOARD
0062    7F      0000G           CLR     BLINK
                                ;******** TURN ON THE RUN LIGHT
```

IDLE MAIN MONITOR IDLE LOOP

```
 58  0135  86  80              LDA A  BSYLT,I        ; SET THE BUSY LAMP
 59  0137  B7  0000G           STA A  PIALT
 60  013A  86  9C              LDA A   H9C,I  ; SET "REN" -- REMOTE ENABLE
 61  013C  B7  0000G           STA A  PIASRQ
 62  013F  96  00G             LDA A  KBFLAG,D
 63  0141  8A  02              ORA A  RUNN,I         ; SET KEYBOARD RUN MODE
 64  0143  97  00G             STA A  KBFLAG,D
 65  0145  85  40              BIT A  ENDKEY,I       ; CR TYPE?
 66  0147  27  05              BEQ    STPK
 67  0149  BD  0000G           JSR    TRANSL         ; GO TRANSLATE A LINE
 68  014C  20  86        IDL.: TRNSL  BRA    IDLE
 69  014E  85  20        STPK: BIT A  STPKEY,I       ; STEP KEY?
 70  0150  27  0E              BEQ    ATLD
 71  0152  D6  00G             LDA B  GLBFLG,D       ; SET STEP FLAG
 72  0154  CA  40              ORA B  STPFLG,I       ; XXXMAY NOT BE NEEDEDXXX
 73  0156  D7  00G             STA B  GLBFLG,D
 74  0158  BD  0000G           JSR    EDTCLS
 75  0158  BD  0000G           JSR    LNEVL          ; DO THE STEP
 76  015E  20  EC              BRA    IDL.           ; AND RESET
 77  0160  85  08        ATLD: BIT A  ATLOD,I        ; ATUO LOAD KEY
 78  0162  27  E8              BEQ    IDL.
 79  0164  BD  0000G           JSR    EDTCLS
 80  0167  BD  0000G           JSR    ATLOAD
 81  016A  20  E0              BRA    IDL.
 82                            .GLOBL ATLOAD
```

```
                                        .SBTTL  KBQOUT   QUE OUTPUT FOR TYPE AHEAD
                                        .GLOBL  KBQOUT
                                ; THIS ROUTINE REMOVES KEYCODES FROM THE KEY Q
                                ; IT MUST NOT BE CALLED IF THE Q IS EMPTY
                                ;  CHAR. RETURNED IN R1
                                ; xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
016C    DE      000     KBQOUT: LDX     QOUT,D          ; GET OUTPUT POINTER
016E    8C      FFFE0           CPX     QEND-2,I        ; NEED TO WRAP AROUND END?
0171    26      05              BNE     TRYOUT
0173    CE      0000G           LDX     KEYQUE,I        ; SET POINTER TO BEGINNING OF QUE
0176    20      01              BRA     TRY.C
0178    08              TRYOUT: INX                     ; MOVE POINTER TO GOOD CHARACTER
0179    A6      00      TRY.C:  LDA A   0,X             ; PICK UP THE CHARACTER
017B    DF      00G             STX     QOUT,D          ; SAVE INDEX REG.
017D    39                      RTS
```

```
                                        .SBTTL  CIDLE CURSOR IDLE LOOP
                                ; THIS ROUTINE DOES AN IDLE WAIT LOOP
                                ; AND ONLY RETURNS UPON ENCOUNTERING AN
                                ; ENDKEY + USER FUNCTION KEY + STEP KEY.  IT
                                ; WILL DISPLAY THE CHARACTER IN CURSOR
                                ; AS THE FLASHING CURSOR IT CALLS THE EDITOR
                                ; AND WHATEVER ELSE IT CAN.
                                ;
                                ; THIS ROUTINE ALSO WAITS FOR ABOUT 30 MINUTES AFTER THE LAST "RETURN" AND
                                ; THEN PAGES THE SCREEN  --  HOW'S THAT FOR SAVING YOUR DUST?
                                ;
017E                            CIDLE:
017E    D6      000G    CIDLE1: LDA B   KBFLAG,D        ; GET KEYBOARD STATUS
0180    C5      68              BIT B   ENDKEY!STPKEY!ATLOD,I   ; TEST IF AN EXIT KEY WAS HIT
0182    27      01              BEQ     1$
0184    39              2$:     RTS                     ; RETURN
0185    96      000G    1$:     LDA A   CRTFLG,D        ; TEST FOR INTERUPTS
0187    9A      000G            ORA A   PNDEOF,D
0189    CE      0000G           LDX     KEYSTK,I        ; CHECK FOR IMMEDIATE EXECUTE KEYS
018C    A6      00              ORA A   0,X
018E    27      19              BEQ     IDLE.C
0190    C5      01              BIT B   INPUTK,I        ; THIS TO MAKE USER DEF. IN INPUT WORK
0192    27      F0              BEQ     2$
0194    96      000G            LDA A   GLBFLG,D
0196    85      10              BIT A   KEYFLG,I
0198    26      0F              BNE     IDLE.C
019A    A6      00              LDA A   0,X             ; IF USER KEY FAKE EOF
019C    27      08              BEQ     IDLE.C
019E    86      0D              LDA A   CR,I
01A0    BD      0082'           JSR     KBPROC          ; GO EDIT IT
01A3    86      08              LDA A   CREOF,I         ; SET EOF FOR INPUT DRIVER
01A5    97      000G            STA A   CRSTAT,D
01A7    20      08              BRA     2$
01A9    DE      000G    IDLE.C: LDX     QIN,D           ; CHECK IF QUE IS EMPTY
01AB    9C      000G            CPX     QOUT,D          ; IF QOUT = QIN THEN EMPTY
01AD    27      08              BEQ     GCURSE
01AF    BD      016C'           JSR     KBQOUT          ; GET A KEY FROM THE QUE
01B2    BD      0082'           JSR     KBPROC          ; PROCESS THE KEY
01B5    20      C7              BRA     CIDLE1
01B7    BD      0000G   GCURSE: JSR     GENCUR          ; DO THE CURSOR.
01BA    20      C2              BRA     CIDLE1
```

```
                                        .SBTTL  KBQIN QUE INPUT FOR TYPE AHEAD
                                        .GLOBL  KBQIN
                                ; THIS ROUTINE PUTS KEY CODES INTO THE KEY Q AFTER
                                ; THEY HAVE PASSED THRU KBINT - THEY KEYBOARD
                                ; INTERRUPT PROCESSOR
                                ; THE KEYCODE IS PASSED IN THE A ACC.
                                ; QIN - POINTER TO NEXT AVAILABEL STORAGE POSITION
                                ; QOUT - POINTER TO NEXT AVAILABLE CHAR
                                ; QSTRT - FIRST POSITION ADDRESS
                                ; QEND - LAST POSITION ADDRESS + 1
                                ; XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;
016C    DE      000     KBQIN:  LDX     QIN,D           ; GET INPUT POINTER
016E    8C      FFFE0           CPX     QEND-2,I        ; NEED TO WRAP AROUND?
0171    26      05              BNE     TRYIN
0173    CE      00000           LDX     KEYQUE,I        ; MOVE POINTER TO BEGINNING OF Q
0176    20      01              BRA     TRY.B
0178    08              TRYIN:  INX                     ; MOVE POINTER TO NEXT SLOT
0179    9C      000     TRY.B:  CPX     QOUT,D          ; QUE FULL?
017B    26      03              BNE     QUEIN
017D    7E      00000           JMP     BELCAL
                                .GLOBL  BELCAL          ; BEEPS THE BELL
0180    A7      00      QUEIN:  STA,A   0,X             ; SAVE THE KEY IN QUE
0182    DF      000             STX     QIN,D
0184    39                      RTS
                0001'           .END
```

SYMBOL TABLE

```
ABRFLG=  0040        ADRDEV= XXXXXX G     AFAIL =  0030        AMAT  =  0010        ARRAY =  0020
ASTR  =  0020        ATLD    0160R        ATLOAD=  XXXXXX G    ATLOD =  0008        ATSNTG=  XXXXXX G
ATVLD =  0004        AUTONO= XXXXXX G     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    BAKSTG=  XXXXXX G
BANK  =  XXXXXX G    BELCAL=  XXXXXX G    BERALC=  XXXXXX G    BERSIT=  0020        BLINK =  XXXXXX G
BMAT  =  0004        BRKCNT=  XXXXXX G    BRKKEY   0068R       BRKKY =  00EB        BRKLT =  0020
BRKSET   0083R       BRK1     0082R       BSTR  =  0008        BSYLT =  0080        BUSACT=  0010
CDOPTR=  XXXXXX G    CDSPTR=  XXXXXX G    CHOR  =  XXXXXX G    CHRCNT=  XXXXXX G    CIDLE    017ERG
CIDLE1   017ER       CLPTR =  XXXXXX G    CLRBNK=  XXXXXX G    CMAT  =  0001        COLCNT=  XXXXXX G
CPY      0021R       CR    =  0000        CRDC3 =  0002        CREOF =  0008        CREOI =  0004
CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT=  XXXXXX G    CRTRST=  XXXXXX G
CRVLD =  0080        CSTR  =  0002        CTKN  =  XXXXXX G    CURSOR=  XXXXXX G    DATDEV=  0022
DIMFLG=  0004        DIRCT =  0080        DISCNT=  XXXXXX G    DISSRQ=  0008        DL    =  XXXXXX G
DP    =  XXXXXX G    DREXTA=  XXXXXX G    DREXTB=  XXXXXX G    DSPCPY=  XXXXXX G    DSPDEV=  0020
DSPSTT=  XXXXXX G    DT    =  XXXXXX G    EDITIT   0085R       EDTBFR=  XXXXXX G    EDTCLS=  XXXXXX G
EDTMAX=  XXXXXX G    EDTPTR=  XXXXXX G    END   =  XXXXXX G    ENDIT    0080R       ENDKEY=  0040
EOETYP=  0038        EOLTG =  XXXXXX G    EOSTG =  XXXXXX G    ERRTSN=  XXXXXX G    ERROMN=  XXXXXX G
EREOM =  XXXXXX G    ERFBFR=  XXXXXX G    ERFILE=  XXXXXX G    ERIOE =  XXXXXX G    ERNIOD=  XXXXXX G
ERNSEP=  XXXXXX G    ERRCD =  XXXXXX G    ERRCDB=  XXXXXX G    ERTERM=  XXXXXX G    ERUNDF=  XXXXXX G
ESTG  =  XXXXXX G    EXTFLG=  0080        FILDEV=  0000        FIX1A =  0003        FIX1B =  0004
FMTVLD=  0008        FNFLG =  0010        FORTC =  XXXXXX G    GCURSE   0187R       GENCUR=  XXXXXX G
GETKEY=  XXXXXX G    GLBFLG=  XXXXXX G    GOSTG =  XXXXXX G    IDLE     0004RG      IDLE.C   01A9R
IDLWCT=  XXXXXX G    IDL      014CR       IECIFC=  XXXXXX G    IECOEF=  XXXXXX G    IMMED    00BAR
IMMEX    0010R       IMXTG =  XXXXXX G    INITMT=  XXXXXX G    INPUT =  XXXXXX G    INPUTK=  0001
IOBFR1=  XXXXXX G    IOFLGS=  XXXXXX G    IOFUNC=  XXXXXX G    IOPLT =  0040        ITM1TG=  XXXXXX G
ITM2TG=  XXXXXX G    LTTDEV=  0024        JMPX  =  XXXXXX G    KBDEV =  001E        KBFLAG=  XXXXXX G
KBIN  =  XXXXXX G    KBINT    0000RG      KBPROC   0082RG      KBQIN    018CRG      KBQOUT   016CRG
KEYFLG=  0010        KEYQUE=  XXXXXX G    KEYSTK=  XXXXXX G    KYRATE=  XXXXXX G    LBRKTG=  XXXXXX G
LCLFLG=  XXXXXX G    LDAX  =  XXXXXX G    LDXX  =  XXXXXX G    LENGTH=  XXXXXX G    LINELN=  XXXXXX G
LISTTG=  XXXXXX G    LNEVL =  XXXXXX G    LNNOTG=  XXXXXX G    LOCTG =  XXXXXX G    LSP   =  XXXXXX G
LSTFMT=  0002        MAXUKY=  00E9        MINUKY=  00C0        MKCPYK=  00ED        MSKCTR=  XXXXXX G
MTBFIN=  XXXXXX G    MTBFR =  XXXXXX G    MTPDEV=  0021        MTPD2 =  0023        MTPRMD   0028RG
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=  0005        NULLTG=  XXXXXX G    OBJATR=  0002        OBJBCK=  0003
OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002        OPRADR=  XXXXXX G    OTHKEY   00C5R
PAETG =  XXXXXX G    PAGEKY=  00EA        PARM  =  0008        PCHAR =  XXXXXX G    PGCTR =  XXXXXX G
PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMFP =  0003        PGMLEN=  0000
PGMLNN=  0007        PGMPTR=  XXXXXX G    PGMTG =  XXXXXX G    PIALT =  XXXXXX G    PIASRQ=  XXXXXX G
PLOSTG=  XXXXXX G    PNDEOF=  XXXXXX G    PNDFLG=  XXXXXX G    PNINTG=  XXXXXX G    PNISTG=  XXXXXX G
POINT =  XXXXXX G    PPMODE=  XXXXXX G    PRIDEF=  0001        PRTERR=  XXXXXX G    PRTTG =  XXXXXX G
PSCTG =  XXXXXX G    QEND  =  XXXXXX G    QIN   =  XXXXXX G    QOUT  =  XXXXXX G    QTRNSL   014CRG
QUEIN    0.D0R       RECLFG=  0004        REALL =  00C0        RMAT  =  0040        RPTCTL=  0080
RST      0018R       RSTKY =  00CA        RSTR  =  0080        RST.1    001ER       RTRNTG=  XXXXXX G
RUNFLG=  0080        RUNN  =  0002        RUNOKY=  00EC        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        SECDEF=  0002
SEMITG=  XXXXXX G    SETBNK=  XXXXXX G    SHIFTK=  0020        SNDIT =  0080        SPCKEY   0061R
STACKK   0053R       STAT37=  XXXXXX G    STEPKY=  00F6        STKIT    0048R       STPFLG=  0040
STPK     014ER       STPKEY=  0020        STRING=  0010        SYSERR=  XXXXXX G    TABPTR=  XXXXXX G
TAGPTR=  XXXXXX G    TCOL  =  XXXXXX G    TRANSL=  XXXXXX G    TRCFLG=  0020        TRYIN    01C8R
TRYOUT   0178R       TRY.B    01C9R       TRY.C    0179R       TSTINT=  XXXXXX G    TYPIN =  XXXXXX G
T51DEV=  0025        UNADP =  XXXXXX G    UNADR3=  XXXXXX G    UNDEF =  0080        USERKY   0040R
```

```
KBPROC KEYBOARD PROCESSOR      RT-11 MMAC  VM02-10   14-OCT-76 01:35:34 PAGE 7+
SYMBOL TABLE

VALTG = xxxxxx G      VALUND=  004G         XAXIS = xxxxxx G      XEQSP = xxxxxx G      XEQSTK= xxxxxx G
ZX    = xxxxxx G
. ABS.    0000     00
          0105     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2278. WORDS
.SY:KBPROC/C'DK1:SE1CLI,KBPROC
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

NTPTR	1-22#							
NTRELY	1-215#							
NTSPTR	1-97#							
NTVBL	1-82#							
NTWCOL	1-91#							
NTWLEN	1-96#							
NTWROW	1-90#							
NULLTG	1-122#							
OBJATR	1-102#							
OBJBCK	1-103#							
OBJDI	1-104#							
OBJLEN	1-101#							
ONSFLG	1-29#							
OPRADR	1-239#	1-329#	4-39%					
OTHKEY	3-19	3-22#						
PAETG	1-133#							
PAGEKY	1-344#	2-21						
PARM	1-85#							
PCHAR	1-313#	2-28						
PGCTR	1-317#							
PGMATR	1-109#							
PGMBP	1-111#							
PGMCD	1-113#							
PGMFP	1-110#							
PGMLEN	1-108#							
PGMLNN	1-112#							
PGMPTR	1-48#							
PGMTG	1-128#							
PIALT	1-337#	2-32	2-35%	2-41%	2-92	2-94%	4-41%	4-59%
PIASRQ	4-9#	4-43%	4-61%					
PLOSTG	1-123#							
PNDEOF	1-45#	6-18						
PNDFLG	1-44#	2-88	2-91%	2-100%	4-51	4-53%	6-17	
PNTNTG	1-132#							
PNTSTG	1-130#							
POINT	1-241#							
PPMODE	1-224#							
PRTDEF	1-206#							
PRTERR	1-312#	4-20						
PRTTG	1-141#							
PSCTG	1-131#							
QEND	1-334#	5-8	7-14					
QIN	1-335#	2-84%	6-34	7-13	7-24%			
QOUT	1-336#	2-83	5-7	5-14%	6-35	7-19		
QTRNSL	4-8#	4-68#						
QUEIN	7-20	7-23#						
R0	1-38#							
R1	1-38#							
R10	1-38#							
R11	1-38#							
R12	1-38#							
R13	1-38#							
R14	1-38#							
R15	1-38#							
R16	1-38#							
R17	1-39#							
R18	1-39#							

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

SE1 1-38

```
KK      KK  EEEEEEEEEE  YY      YY  DDDDDDDDD   RRRRRRRRR   VV      VV          LL          SSSSSSSS   TTTTTTTTTT
KK      KK  EEEEEEEEEE  YY      YY  DDDDDDDDDD  RRRRRRRRRR  VV      VV          LL          SSSSSSSSSS  TTTTTTTTTT
KK   KK     EE          YY  YY      DD      DD  RR      RR  VV      VV          LL          SS      S       TT
KK KK       EE            YYYY      DD      DD  RR      RR  VV      VV          LL          SS              TT
KKKKK       EEEEEE         YY       DD      DD  RRRRRRRRRR  VV      VV          LL          SSSSSSSS        TT
KK  KK      EEEEEE         YY       DD      DD  RRRRRRRRR   VV      VV          LL          SSSSSSSS        TT
KK   KK     EE             YY       DD      DD  RR   RR     VV    VV           LL                SS        TT
KK    KK    EE             YY       DD      DD  RR    RR    VV  VV            LL          S     SS        TT
KK     KK   EEEEEEEEEE     YY       DDDDDDDDDD  RR     RR    VVVV     ......  LLLLLLLLLL  SSSSSSSSS       TT
KK      KK  EEEEEEEEEE     YY       DDDDDDDDD   RR      RR   VV       ....    LLLLLLLLLL  SSSSSSSS        TT
```

KEYDRV--KEYCODE PROCESSOR RT-11 MNAC VM02-10 14-OCT-76 01:35:50
TABLE OF CONTENTS

```
                                        .TITLE  KEYDRV--KEYCODE PROCESSOR
                                        .IDENT  /SBR009/
                                        .SBTTL  CONVERT MATRIX CODES TO ASCII
                                ;*****************************************************************
                                ;THIS SECTION READS THE SWITCH MATRIX AND CONVERTS THE KEYS TO
                                ;ASCII OR SPECIAL CODES
                                ;
                                ;SPECIAL CODES HAVE BIT 7=1
                                ;       SPECIAL CODE=80 MEANS NO VALID KEY
                                ;ASCII CODES HAVE BIT 7=0
                                ;
                                        .GLOBL  GETKEY
                                        .GLOBL  KEYDRV
0000'                           KEYDRV  =.
0000                            GETKEY:
                                        .GLOBL  CNTL,TTY2,SHIFT1,SHIFT2,ROW,COL,STKUSE,KEYTIM
                                        .GLOBL  KBIN, NOKEY
                                        .GLOBL  PGCTR
                                        .GLOBL  KEYCNT,KBSTK,KBMRK,ACCEL
                                        .GLOBL  PIAKB,PIALT
                                        .GLOBL  KYRATE,VLOKEY
                                ;*****************************************************************
                                ;READ SWITCH MATRIX, SHIFT KEYS, AND TTY LOCK
                                ;CONVERT MATRIX CODE TO ROW AND COL NUMBERS
                                ;FORM PROPER SHIFT FLAGS AND BYTES
                                ;
0000  CE  3348                          LDX     13131.,I        ; SET PAGE COUNTER
0003  FF  0000G                         STX     PGCTR
0006  B6  0000G                         LDA A   PIAKB           ;READ MATRIX CODE
0009  16                                TAB
000A  84  7F                            AND A   H7F.I   ;MASK OUT TTY LOCK
000C  26  19                            BNE     RELKEY          ;IF ZERO THEN NO KEY IS DOWN
                                ;*****************************************************************
                                ;INITIALIZE SOME FLAGS AND BYTES
                                ;
000E  7F  0000G                         CLR     STKUSE          ;CLEAR THE KEY STACK
0011  7F  0000G                         CLR     KEYTIM          ; CLEAR KEY TIMER
0014  86  FF                            LDA A   HOFF.I  ;CONTROL KEY MASK BYTE
0016  B7  0000G                         STA A   CNTL
0019  7F  0000G                         CLR     VLOKEY
001C  7F  0000G                         CLR     TTY2            ;TTY LOCK CONSTANT
001F  7F  0000G                         CLR     SHIFT1          ;SHIFT CONSTANT 1
0022  7F  0000G                         CLR     SHIFT2          ;SHIFT CONSTANT 2
0025  20  4E                            BRA     NOKEYX          ; AND EXIT
0027  97  00G           RELKEY: STA A   KBIN,D          ; SAVE 7 BIT CODE
0029  7D  0000G                         TST     STKUSE
002C  26  26                            BNE     GO
002E  5D                                TST B
002F  2B  05                            BMI     1$
0031  C6  20                            LDA B   H20.I
0033  F7  0000G                         STA B   TTY2
0036  B6  0000G         1$:     LDA A   PIALT           ;READ CONTROL AND SHIFT KEYS
0039  88  03                            EOR A   3,I
0038  85  03                            BIT A   3,I             ;SHIFT KEY DEPRESSED
003D  27  15                            BEQ     GO
003F  C6  10                            LDA B   H10.I   ;SET UP SHIFT CODES
0041  F7  0000G                         STA B   SHIFT1
```

```
 331  0044  58                    F[illegible]
 332  0045  F7  00000G            [illegible]  SHIFT?
 333  0048  F7  00000G            STA  TTY2          ;SET TTY LOCK AS WELL
 334  004B  85  02                BIT  A,1           ;CONTROL KEY DEPRESSED?
 335  004D  27  06                BEQ  GO
 336  004F  C6  1F                LDA B  #01F,I ;SET UP CONTROL KEY CODE
 337  0051  F7  00000G            STA B  CNTL
 338  0054                  GO:
```

```
                                .SBTTL  DEBOUNCE AND REPEAT PROCESSOR
                                ;*****************************************************************
                                ; DEBOUNCE, REPEAT AND 4 KEY CALL OVER LOGIC
                                ; ENTER WITH ASCII FORM KEYCODE IN KBIN (TEMPORARY STORAGE).
                                ; KEYS ARE STORED AS A FIRST IN FIRST OUT STACK
                                ; IN THE FOLLOWING MANNER.
                                ; KBSTK:        KEY0 - FIRST KEY
                                ;               MARK0 - 1 IF ACCESSED THIS SCAN
                                ;               KEY1
                                ;               MARK1
                                ; UP THRU        MARK3
                                ; KEYCNT = NUMBER OF TIMES/2 THAT KEY0 HAS BEEN SCANNED
                                ;          SINCE LAST TIME IT WAS TRULY VALID.  THIS IS FOR
                                ;          REPEAT FUNCTION
                                ; KEYTIM = WHAT KEYCNT MUST BE BEFORE ISSUING NEXT VALID
                                ;          KEY CODE.
                                ;*****************************************************************
                                ; TEST IF KEY IS ALREADY ON KEYSTACK
0054    CE    0000G                     LDX     KBSTK,I         ; GET START OF KEY STACK
0057    96    00G                       LDA A   KBIN,D          ; GET NEWKEYCODE
0059    F6    0000G                     LDA B   STKUSE          ; STACK USAGE COUNT, ZERO IF STACK EMPTY
005C    27    09        STKTST: BEQ     NEWKEY          ; TRY TO STACK NEW KEY
005E    A1    00                CMP A   0,X             ; KEY ON STACK?
0060    27    18                BEQ     FNDKEY          ; PROCESS IT IF FOUND
0062    08                      INX                     ; MOVE TO NEXT ENTRY
0063    08                      INX
0064    5A                      DEC B
0065    20    F5                BRA     STKTST          ; TRY AGAIN
                                ;*****************************************************************
                                ; TRY TO ENTER NEW KEY CODE ONTO THE STACK
0067    F6    0000G     NEWKEY: LDA B   STKUSE          ; GET USAGE COUNT
006A    C1    07                CMP B   7,1             ; STACK FULL?
006C    2C    07                BGE     NOKEYX          ; IF FULL EXIT WITH NO KEY
006E    7C    0000G             INC     STKUSE          ; INC. STACK USAGE
0071    A7    00                STA A   0,X             ; STORE NEW KEY ON STACK
0073    6F    01                CLR     1,X             ; CLEAR MARK
0075    86    80        NOKEYX: LDA A   NOKEY,I         ; EXIT WITH NO KEY CODE
0077    97    00G               STA A   KBIN,D
0079    39                      RTS
                                ;*****************************************************************
                                ; FOUND KEY ON STACK
                                ; TEST IF IT IS ALREADY MARKED  (1) IF SO THEN HAVE
                                ; COMPLETED ONE SCAN - NEED TO CLEAN STACK, (2) ELSE
                                ; JUST MARK IT
007A    A6    01        FNDKEY: LDA A   1,X             ; TEST MARK
007C    81    02                CMP A   2,1
007E    24    09                BCC     GOODKY
0080    6C    01                INC     1,X             ; ELSE JUST MARK IT
0082    7D    0001G             TST     KBSTK+1         ; LOOK AT TOP ENTRY
0085    27    FF                BEQ     NOKEYX          ; LOOK FOR LOWEST KEY ON STACK
0087    20    1F                BRA     PRSKEY
                                ;*****************************************************************
                                ; GOOD KEY FOUND (MARKED TWICE), SCAN COMPLETE
                                ; 1) CLEAR ANY KEYS FROM STACK ABOVE THE GOOD KEY
                                ;2) CLEAR MARK ON ANY OF THE REMAINING KEYS
0089    FF    0000G     GOODKY: STX     KBMRK           ; SAVE POSITION OF NEW KEY
008C    CE    0000G             LDX     KBSTK,I         ; GET START OF STACK
```

```
008F   BC  0000G          CPX    KBMRK       ; NEW KEY AT TOP OF STACK YET?
0092   27    14           BEQ    PRSKEY      ; THEN GO PROCESS THE KEY
0094   7A  0000G          DEC    STKUSE      ; DEC. STACK USAGE COUNTER
0097   C6    06           LDA B  6,I         ; COUNTER TO MOVE STACK UP ONE ENTRY
0099   A6    02   DELOOP: LDA A  2,X         ; THIS MOVES HALF AN ENTRY
009B   A7    00           STA A  0,X
009D   08                 INX
009E   5A                 DEC B
009F   26    F8           BNE    DELOOP      ; DONE MOVING THE STACK?
00A1   FE  0000G          LDX    KBMRK       ; GET NEW KEY MARKER
00A4   09                 DEX                ; MOVE IT BACK ON ENTRY
00A5   09                 DEX
00A6   20    E1           BRA    GOODKY      ; GO TRY AND CLEAR SOME MORE
                  ; ******************************************************
                  ; PROCESS KEY
                  ; 1) HERE WHEN TOP ENTRY IS MARKED
                  ; 2) LOOK FOR MARKED ENTRY NEAREST BOTTOM OF STACK
                  ; 3) ZERO ALL MARKS
                  ;
00A8   CE  0000G  PRSKEY: LDX    KBSTK,I
00AB   B6  0000G          LDA A  STKUSE
00AE   E6    00   1$:     LDA B  0,X
00B0   4A         2$:     DEC A
00B1   27    08           BEQ    FNDONE
00B3   08                 INX
00B4   08                 INX
00B5   6D    01           TST    1,X
00B7   26    F5           BNE    1$
00B9   20    F5           BRA    2$
00BB   CE  0000G  FNDONE: LDX    KBSTK,I
00BE   D7    00G          STA B  KBIN,D      ; SAVE KEY FOR LATER
00C0   6F    01           CLR    1,X
00C2   6F    03           CLR    3,X         ; CLEAR MARK ALL ENTIES
00C4   6F    05           CLR    5,X
                  ;       CLR    7,X
00C6   F1  0000G          CMP B  VLDKEY
00C9   27    0C           BEQ    SECOND      ;IS THIS THE FIRST TIME FOR THIS KEY
00CB   86    06           LDA A  6,,I        ;SET ACCELERATION
00CD   B7  0000G          STA A  ACCEL
00D0   86    37           LDA A  55.,I       ;SET INITAL COUTN TIME
00D2   B7  0000G          STA A  KEYTIM
00D5   20    2A           BRA    SETTIM      ;SET  COUNTERS AND EXIT
00D7   7A  0000G  SECOND: DEC    KEYCNT      ;DECREMENT KEY DOWN TIME COUNTER
00DA   26    99           BNE    NOKEYX      ;EXIT IF NOT DONE YET
00DC   D6    00G          LDA B  KBIN,D      ;IS IT BREAK KEY WHICH DOESN'T REPEAT
00DE   C1    6B           CMP B  H06B,I ;
00E0   27    93           BEQ    NOKEYX
00E2   B6  0000G          LDA A  KEYTIM      ; GET PRESENT KEYTIME
00E5   81    37           CMP A  55.,I       ;DONE INITIAL DELAY?
00E7   26    05           BNE    NEWTIM
00E9   86    1A           LDA A  26.,I
00EB   B7  0000G          STA A  KEYTIM
00EE   B1  0000G  NEWTIM: CMP A  KYRATE      ;ALREADY RUNNING AT FAST RATE?
00F1   23    0E           BLS    SETTIM      ;RRESET TIMES
00F3   B0  0000G          SUB A  ACCEL       ;ELSE DO ACCELERATION
00F6   B7  0000G          STA A  KEYTIM
00F9   7A  0000G          DEC    ACCEL       ;DECREMENT ACCELERATION TIME
```

```
115  00FC  26  03          BNE   SETTIM
116  00FE  7C  0000G       INC   ACCEL          ;MAKE SURE IT'S NOT ZERO
117  0101  B7  0000G SETTIM: STA P KEYCNT       ;SET REAL KEY COUNTER
118  0104  96  00G         LDA   KBIN,D
119  0106  B7  0000G       "     VLDKEY         ; SAVE THIS KEYCODE FOR LATER
120  0109  16                                   ;FORM ROW NUMBER
121  010A  C4  0F          AND B  #0F,I
122  010C  F7  0000G       STA B  ROW
123  010F  47              ASR A                ;FORM COLUMN NUMBER
124  0110  47              ASR A
125  0111  47              ASR A
126  0112  47              ASR A
127  0113  B7  0000G       STA A  COL
128  0116  D6  00G         LDA B  KBIN,D        ; GET FULL MATRIX CODE
129  0118  CE  012C'       LDX    COLCDE,I      ; ROUTINE ADDRESSES
130  0118  B6  0000G       LDA A  COL           ; GET COL. NUMBER
131  011E  27  05        2$:BEQ    1$           ; FOUND PROPER COLUMN YET?
132  0120  08              INX
133  0121  08              INX
134  0122  4A              DEC A
135  0123  20  F9          BRA    2$           ; TRY AGAIN
136  0125  EE  00        1$:LDX    0,X          ; GET ROUITINE ADDRESS
137  0127  AD  00          JSR    0,X
138  0129  D7  00G         STA B  KBIN,D        ; SAVE THE REAL CODE
139  012B  39              RTS
```

```
                                .SBTTL  COLUMN PROCESSOR CODE
                                ;**************************************************************************
                                ;DISPATCH TO COLUMN PROCESSOR TABLE
                                ;
012C    013C'                   COLCDE: COL0                    ;COLUMN 0
012E    013F'                           COL1
0130    0147'                           COL2
0132    0151'                           COL3
0134    0164'                           COL4
0136    017A'                           COL5
0138    019A'                           COL6
013A    01A7'                           COL7
                                ;**************************************************************************
                                ;THE COLUMN CODE ROUTINES
                                ;
                                ;COLUMN 0
                                ;
013C    CB      80              COL0:   ADD B   #80.,1  ; TURN ON MSB FOR WORDY KEYS
013E    39                              RTS                     ;ROUGH YES
                                ;
                                ;COLUMN 1
                                ;
013F    7D      00000G          COL1:   TST     ROW             ; IF ROW ZERO MAKE IT SPACE
0142    26      02                      BNE     1$
0144    CB      10                      ADD B   16.,1
0146    39              1$:     RTS
                                ;
0147    B6      00000G          COL2:   LDA A   ROW             ;GET ROW NUMBER
014A    81      0A                      CMP A   #0A.,1  ;ROWS A-F ARE OK
014C    24      02                      BCC     1$
014E    CB      10                      ADD B   #10.,1  ;ROWS 0-9 NEED THIS
0150    39              1$:     RTS
                                ;
                                ;COLUMN 3
                                ;
0151    B6      00000G          COL3:   LDA A   ROW             ;WORK ON ROW
0154    27      0D                      BEQ     RTN3            ;ROW 0?
0156    81      08                      CMP A   #08.,1
0158    22      04                      BHI     1$
015A    F0      00000G                  SUB B   SHIFT1          ;ROW 1-8 MODIFICATION
015E    39                              RTS                     ;RETURN
015E    C0      10              1$:     SUB B   #10.,1  ;ROW C-F
0160    FB      00000G                  ADD B   SHIFT1          ;ADD IN SHIFT KEY INFO
0163    39              RTN3:   RTS                     ;RETURN
                                ;
                                ;COLUMN 4 PROCESSOR
                                ;
0164    B6      00000G          COL4:   LDA A   ROW             ;WORK ON ROW AGAIN
0167    26      08                      BNE     COL4.1          ; ROW 0?
0169    FB      00000G                  ADD B   SHIFT2          ;TAKE CARE OF SHIFT KEY
016C    F4      00000G                  AND B   CNTL
016F    20      08                      BRA     RTN4
0171    CB      20              COL4.1: ADD B   #20.,1  ;ROWS 1-F
0173    F0      00000G                  SUB B   TTY2            ;DO SHIFT KEY
0176    F4      00000G          COL4.M: AND B   CNTL            ;DO CONTROL KEY
0179    39              RTN4:   RTS                     ;RETURN
                                ;
```

COLUMN PROCESSOR CODE

```
                                        ;COLUMN 5
                                        ;
017A  B6 0000G   COL5:   LDA A  ROW             ;GET ROW NUMBER
017D  81 0A              CMP A  #0A,I
017F  22 07              BHI    COL5.B
0181  CB 20              ADD B  #20,I   ;ROW 0-A
0183  F0 0000G           SUB B  TTY2            ;TAKE CARE OF SHIFT
0186  20 0E              BRA    COL5.M          ;GO DO CONTROL
0188  C1 0E      COL5.B: CMP A  #0E,I
018A  22 05              BHI    COL5.F
018C  FB 0000G           ADD B  SHIFT2          ;ROW B-E
018F  20 05              BRA    COL5.M          ;GO DO CONTROL
0191  CB 20      COL5.F: ADD B  #20,I   ;ROW F
0193  F0 0000G           SUB B  SHIFT2
0196  F4 0000G   COL5.M: AND B  CNTL            ;DO CONTROL MASK
0199  39                RTS                    ;RETURN
                                        ;
                                        ;COLUMN 6
                                        ;
019B  CB 80      COL6:   ADD B  #80,I   ;ALL ROWS NEED THIS
019C  B6 0000G           LDA A  ROW
019F  81 0A              CMP A  #0A,I
01A1  22 03              BHI    1$
01A3  F0 0000G           SUB B  SHIFT2          ;ROW 0-A NEED TO BE SHIFTABLE
01A6  39         1$:     RTS                    ;RETURN
                                        ;
                                        ;COLUMN 7
                                        ;
01A7  B6 0000G   COL7:   LDA A  ROW             ; GET ROW
01AA  80 0E              SUB A  #0B,I  ; IS IT ROW >=C?
01AC  22 0F              BHI    FNYKEY          ; THEN IT IS A FUNNY KEY
01AE  CB 80              ADD B  #80,I  ; IMMEDIATE EXECUTE KEYS NEED THIS
0180  8B 06              ADD A  6,I
0182  2B 01              BMI    1$
0184  39                 RTS
0185  17        1$:     TBA                    ; TO MAKE EDIT KEYS SHIFTABLE
0186  F6 0000G            LDA B  SHIFT2         ; IF SHIFT THEN 80-85 ARE EDIT
0189  58                 ASL B
018A  10                 SBA
018B  16                 TAB                    ; PUT IT WHERE IT BELONGS
018C  39                 RTS
018D  CE 01C9'   FNYKEY: LDX    FNYTBL,I       ; GET TABLE
01C0  4A         2$:     DEC A                  ; FOUND KEY?
01C1  27 03              BEQ    1$
01C3  08                 INX
01C4  20 FA              BRA    2$
01C6  E6 00      1$:     LDA B  0,X             ; GET PROPER KEY CODE
01C8  39                 RTS
01C9  5E 28 29 45 FNYTBL: .BYTE  H5E, H28, H29, H45      ;  ,(,),E RESPECTIVELY
01CC
                         .END
```

```
KEYDRV--KEYCODE PROCESSOR      RT-11 MMAC  VM02-10   14-OCT-76 01:35:50 PAGE 3+
SYMBOL TABLE

ABRFLG=  0040      ACCEL = XXXXXX G    AFAIL =  0030      AMAT  =  0010      ARRAY =  0020
ASTR  =  0020      ATLOD =  0008       ATSNTG= XXXXXX G   ATXLD =  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   BAKSTG= 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 C    CHRCNT= XXXXXX G   CLPTR = XXXXXX G   CMAT  =  0001
CNTL  = XXXXXX G   COL   = XXXXXX G    COLCDE   012CR     COLCNT= XXXXXX G   COLO     013CR
COL1     013ER     COL2     0147R      COL3     0151R     COL4     016AR     COL4.M   0176R
COL4.1   0171R     COL5     017AR      COL5.B   0188R     COL5.F   0191R     COL5.M   0196R
COL6     019AR     COL7     01A7R      CRDC3 =  0002      CREOF =  0008      CREOI =  0004
CREOT =  0020      CRETX =  0010       CRNORM=  0001      CRSTAT= XXXXXX G   CRXLD =  0080
CSTK  =  0002      CTKN  = XXXXXX G    CURSOR= XXXXXX G   DATDEV=  0022      DELOOP   0099R
DIMFLG=  0004      DIRCT =  0080       DISCNT= XXXXXX G   DISSRQ=  0008      DL    = XXXXXX G
DP    = XXXXXX G   DREXTA= XXXXXX G    DREXTB= XXXXXX G   DSPDEV=  0020      DT    = XXXXXX G
EDTBFR= XXXXXX G   ENDKEY=  0040       EOFTYP=  0038      EOLTG = XXXXXX G   EOSTG = XXXXXX G
ERATSN= XXXXXX G   ERDOMN= XXXXXX G    EREOM = XXXXXX G   ERFBFR= XXXXXX G   ERFILE= XXXXXX G
ERIOE = XXXXXX G   ERNLOG= XXXXXX G    ERNSEP= XXXXXX G   ERRCD = XXXXXX G   ERTERM= XXXXXX G
ERUNDF= XXXXXX G   ESTG  = XXXXXX G    EXTFLG=  0080      FILDEV=  0000      FIX1A =  0003
FIX1B =  0004      FMTVLD=  0038       FNDKEY   007AR     FNDONE   008BR     FNFLG =  0010
FNYKEY   018DR     FNYTBL   01C9R      FORTG = XXXXXX G   GETKEY   0000RG    GLBFLG= XXXXXX G
GO       0054R     GOODKY   0089R      GOSTG = XXXXXX G   IMXTG = XXXXXX G   INPUTK=  0001
IOBFR1= XXXXXX G   IOFLGS= XXXXXX G    IOFUNC= XXXXXX G   ITM1TG= XXXXXX G   ITM2TG= XXXXXX G
ITTDEV=  0024      JMPX  = XXXXXX G    KBDEV =  001F      KBFLAG= XXXXXX G   KBIN  = XXXXXX G
KBMRK = XXXXXX G   KBSTK = XXXXXX G    KEYCNT= XXXXXX G   KEYMRV=  0000RG    KEYFLG=  0010
KEYSTK= XXXXXX G   KEYTIM= XXXXXX G    KYRATE= XXXXXX G   LBRKTG= XXXXXX G   LCLFLG= XXXXXX G
LDAX  = XXXXXX G   LDXX  = XXXXXX G    LENGTH= XXXXXX G   LSTTG = XXXXXX G   LNNOTG= XXXXXX G
LOCTG = XXXXXX G   LSP   = XXXXXX G    LSTFMT=  0002      MTBFR = XXXXXX G   MTPDEV=  0021
MTPD2 =  0023      NEWKEY   0067R      NEWTIM   00EER     NLPTR = XXXXXX G   NOKEY =  0080 G
NOKEYX   0075R     NOOUT = XXXXXX G    NOWAIT=  0001      NTAPTR=  0008      NTATTR=  0004
NTDINS=  0009      NTDLEN=  0005       NTLINK=  0000      NTNAME=  0002      NTPTR = XXXXXX G
NTRELY=  0010      NTSPTR=  0008       NTVAL =  0005      NTWCOL=  0007      NTWLEN=  0007
NTWROW=  0006      NULLTG= XXXXXX G    OBJATR=  0002      OBJBCK=  0003      OBJDT =  0005
OBJLEN=  0000      ONSFLG=  0002       OPRADR= XXXXXX G   PAETG = XXXXXX G   PARM  =  0008
PGCTR = XXXXXX G   PGMATR=  0002       PGMBP =  0005      PGMCD =  0009      PGMFP =  0003
PGMLEN=  0000      PGMLNK=  0007       PGMPTR= XXXXXX G   PGMTG = XXXXXX G   PIAR8 = XXXXXX G
PIALT = XXXXXX G   PLOSTG= XXXXXX G    PNDEOF= XXXXXX G   PNDFLG= XXXXXX G   PNTNTG= XXXXXX G
PNTSTG= XXXXXX G   POINT = XXXXXX G    PPMODE= XXXXXX G   PRIDEF=  0001      PRSKEY   00A8R
PRTTG = XXXXXX G   PSCTG = XXXXXX G    RECLEG=  0004      RELKEY   0027R     REAIL =  00C0
RMAT  =  0040      ROW   = XXXXXX G    RPTCTL=  0080      RSTR  =  0080      RTN3     016JR
RTN4     0179R     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
SECDEF=  0002      SECOND   0007R      SEMITG= XXXXXX G   SETTIM   C101R     SHIFT1= XXXXXX G
SHIFT2= XXXXXX G   SNDIT =  0080       STAT37= XXXXXX G   STKIST   005CR     STKUSE= XXXXXX G
STPFLG=  0040      STPKEY=  0020       STRING=  0010      SYSERR= XXXXXX G   TABPTR= XXXXXX G
TAGPTR= XXXXXX G   TCOL  = XXXXXX G    TRCFLG=  0020      TTY2  = XXXXXX G   T51DEV=  0025
UNDEF =  0080      VALTG = XXXXXX G    VALUND=  0040      VLDKEY= XXXXXX G   XAXIS = XXXXXX G

. ABS.   0000      00
         01CD      01

ERRORS DETECTED:  0      WARNINGS POSTED:  0      FREE CORE:  2439. WORDS
,SY:KEYDRV/C:DK1:SETCLI,KEYDRV
```

CROSS REFERENCE TABLE (CREF V01-03.)

1-287

A.END 1-196#
A.MAX 1-197#
A.PRIM 1-192#
A.PTR 1-194#
A.SEC 1-193#
A.STAT 1-191#
A.STRT 1-195#
ABRFLG 1-26#
ACCEL 1-292# 2-96# 2-112 2-114# 2-116#
AFAIL 1-155#
AMAT 1-154#
APRAY 1-87#
ASTR 1-153#
ATLOD 1-216#
ATSNTG 1-138#
ATVLD 1-204#
BAKSTG 1-146#
BANK 1-250#
BFRSTT 1-201#
BLINK 1-222#
BMAT 1-157#
BRKCNT 1-73#
BSTR 1-156#
BUSPCT 1-202#
CDOPTR 1-71#
CDSPTR 1-70#
CHAR 1-189#
CHRCNT 1-242#
CLPTR 1-20#
CMAT 1-159#
CNTL 1-289# 1-312# 1-337# 3-51 3-55 3-72
COL 1-289# 2-127# 2-130
COL0 3-5 3-18#
COL1 3-6 3-23#
COL2 3-7 3-28#
COL3 3-8 3-36#
COL4 3-9 3-48#
COL4.1 3-49 3-53#
COL4.M 3-55#
COL5 3-10 3-60#
COL5.B 3-62 3-66#
COL5.F 3-67 3-70#
COL5.M 3-65 3-69 3-72#
COL6 3-11 3-77#
COL7 3-12 3-86#
COLCDE 2-129 3-5#
COLCNT 1-244#
CRDC3 1-181#
CREOF 1-183#
CREOI 1-182#
CREOT 1-185#
CRETX 1-184#
CRNORM 1-180#
CRSTAT 1-178#
CRVLD 1-187#

CROSS REFERENCE TABLE (CREF V01-03)

```
CSTR    1-158#
CTKN    1-23#
CURSOR  1-223#
DATDEV  1-258#
DELOOP  2-62#   2-66
DIMFLG  1-28#
DIRCT   1-200#
DISCNT  1-221#
DISSRQ  1-36#
DL      1-233#
DP      1-232#
DREXTA  1-271#
DREXTB  1-272#
DSPDEV  1-256#
DT      1-234#
EDTBFR  1-265#
ENDKEY  1-213#
EOFTYP  1-186#
EOLTG   1-142#
EOSTG   1-143#
ERATSN  1-170#
ERDOMN  1-164#
EREOM   1-171#
ERFBFR  1-167#
ERFILE  1-169#
ERIOE   1-168#
ERNIOD  1-173#
ERNSEP  1-165#
ERRCD   1-42#
ERTERM  1-166#
ERUNDF  1-172#
ESTG    1-124#
EXTFLG  1-25#
FILDEV  1-254#
FIX1A   1-117#
FIX1B   1-118#
FMTVLD  1-203#
FNDKEY  2-24    2-45#
FNDONE  2-81    2-87#
FNFLG   1-27#
FNYKEY  3-88    3-99#
FNYTBL  3-99    3-106#
FORTG   1-126#
GETKEY  1-285#  1-288#
GLBFLG  1-31#
GO      1-320   1-328   1-335   1-338#
GOODKY  2-47    2-56#   2-70
GOSTG   1-125#
IMXTG   1-129#
INPUTK  1-219#
IOBFR1  1-266#
IOFLGS  1-246#
IOFUNC  1-190#
ITM1TG  1-136#
ITM2TG  1-137#
ITYDEV  1-260#
JMPX    1-55#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

SEI 1-3#

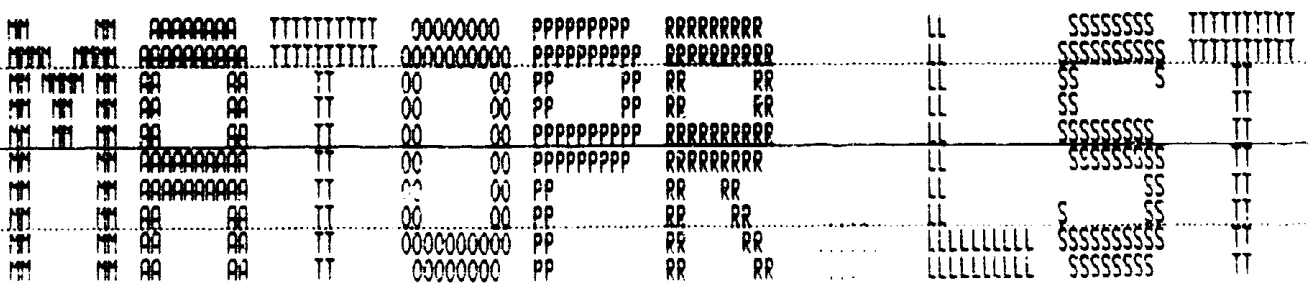


TABLE OF CONTENTS

```
                                .TITLE  MATOPR  SCALER OVER MATRIX OPERATIONS
                                .IDENT  /BE9030/
                        ;
                                .GLOBL  MATOPR
0000'                   MATOPR  =.
                                .GLOBL  SNGMAT,MATMAT,MATSCL,SCLMAT
                                .GLOBL  A5X,A6X,A8X,A9X,A10X,A11X,A12X,A13X
                                .GLOBL  PSHFPN,PULFPN,FPZERO,FPONE,FPADD,ASSCSC
                                .GLOBL  HALTA,ZX
                                .GLOBL  REPLIT,FPALIT,FPC.TESTCS
                                .GLOBL  SETERR,SETERA,OPRCL
                        ;
                        ;       REGISTER USAGE
                        ;
                        ;       R0 - ROW LENGTH AT SETUP TIME
                        ;       R1 - COLUMN LENGTH AT SETUP TIME
                        ;       R2 - WORK REG AT SETUP TIME
                        ;       R5 - NOT USED
                        ;       R6 - CURRENT OUTPUT ADDRESS
                        ;       R7 - STOP OUTPUT ADDRESS
                        ;       R8 - LEFT ARG CRNT ADDR (IF = 0 THIS IS SNGMAT)
                        ;       R9 - LEFT ARG RESET VALUE (IF = 0 NO RESET SHOULD HAPPEN)
                        ;       R10 - RIGHT ARG CRNT ADDR
                        ;       R11 - RIGHT ARG RESET (IF = 0 NO RESET SHOULD HAPPEN)
                        ;
                        ;       OPRADR - THE ADDRESS OF THE OPERATOR ROUTINE TO CALL
                        ;
                        ;
                        ;       MAT=MOP MAT
                        ;
0000    30              SNGMAT: TSX                     ;SET UP OUTPUT AREA CONTROLS
0001    09                      DEX
0002    09                      DEX
0003    BD  0072'               JSR     OUTONE          ;FAKE ENTRY POINT TO RESULT
0006    BD  0093'               JSR     RIGHT           ;SET UP FOR RIGHT ARG
0009    DE  000G                LDX     ZX,0            ;SET UP FOR NO LEFT ARG
000B    DF  000G                STX     R8,0
000D    DF  000G                STX     R9,0
000F    20  27                  BRA     LOOP            ;GO JOIN COMMON ROUTINES
                        ;
                        ;       MAT=MAT DOP MAT
                        ;
0011    BD  006E'       MATMAT: JSR     RESULT
0014    BD  0093'               JSR     RIGHT
0017    BD  009E'               JSR     LEFT
001A    20  1C                  BRA     LOOP
                        ;
                        ;       MAT=MAT DOP SCL
                        ;
001C    BD  006E'       MATSCL: JSR     RESULT
001F    BD  009E'               JSR     LEFT
0022    30                      TSX                     ;SET UP SCALER RIGHT ARG
0023    08                      INX
0024    DF  000G                STX     R10,0
```

```
                                        STX     R11,D
                                        BRA     LOOP
                        ;
                        ;       MAT=SCL OOP MAT
                        ;
002A BD 006E'   SCLMAT: JSR     RESULT
0020 BD 0093'           JSR     RIGHT
0030 30                 TSX                     ;SET UP FOR SCALER LEFT ARG
0031 BD 0000G           JSR     A10X
0034 DF 00G             STX     R8,D
0036 DF 00G             STX     R9,D
                        ;
                        ;       COMMON CODE FOR ALL ROUTINES
                        ;
0038 DE 00G     LOOP:   LDX     R8,D            ;PUT PUT LEFT OPERAND ON STACK
003A 27 0E              BEQ     SKIPA           ;NO SUCH OPERAND
003C BD 0000G           JSR     PSHEPN
003F BD 0000G           JSR     A8X
0042 DF 00G             STX     R8,D
0044 DE 00G             LDX     R9,D            ;DO I NEED TO RESET
0046 27 02              BEQ     SKIPA           ;NO
0048 DF 00G             STX     R8,D
004A DE 00G     SKIPA:  LDX     R10,D           ;NOW GET RIGHT OPERAND
004C BD 0000G           JSR     PSHEPN          ;STACK IT
004F BD 0000G           JSR     A8X
0052 DF 00G             STX     R10,D
0054 DE 00G             LDX     R11,D           ;IS IT A SCALER
0056 27 02              BEQ     SKIPB           ;NO
0058 DF 00G             STX     R10,D
005A BD 0000G   SKIPB:  JSR     OPRCL           ;GENERAL PURPOSE CALL
005D DE 00G             LDX     R6,D
005F BD 0000G           JSR     PULEPN
0062 BD 0000G           JSR     A8X
0065 DF 00G             STX     R6,D
0067 9C 00G             CPX     R7,D
0069 26 CD              BNE     LOOP
0068 7E 0000G           JMP     HALTA           ;SOO LONG
                        ;
                        ;       RESULT TEST AND SETUP SUBROUTINE
                        ;
                        ;       SETS UP REGISTERS AS FOLLOWS
                        ;
                        ;       R6 - ADDRESS OF DATA SPACE
                        ;       R7 - LENGTH OF DATA SPACE (MUST BE RELOCATED FOR MAIN LOOP)
                        ;       R0 - NUMBER OF ROWS FOR SHAPE TEST
                        ;       R1 - NUMBER OF CLOUMNS FOR SHAPE TEST
                        ;
006E 30         RESULT: TSX                     ;GET ADDR OF NT ENTRY
006F BD 0000G           JSR     A9X
0072 DF 00G     OUTONE: STX     R2,D            ;SAVE STACK ENTRY ADDR
0074 EE 0C              LDX     12.,X           ;OUTONE IS FOR ONE OPERAND ENTRY
0076 EE 08              LDX     NTAPTR,X        ;GET DATA ADDR
0078 BD 0000G           JSR     A5X             ;SKIP HEADER
007B DF 00G             STX     R6,D
007D DE 00G             LDX     R2,D
007F EE 0E              LDX     14.,X           ;ADDR OF END OF DATA-5
0081 BD 0000G           JSR     A5X
```

```
                                ;BYTES FOR F.P. NUMBERS NOW
0084  DF  00G       STX   R7,D
0086  DE  00G       LDX   R2,D    ;SET UP SHAPE CONTROLS
0088  EE  10        LDX   16.,X
008A  DF  00G       STX   R0,D
008C  DE  00G       LDX   R2,D
008E  EE  12        LDX   18.,X
0090  DF  00G       STX   R1,D
0092  39           RTS           ;ALL DONE
                    ;
                    ;   OPERAND TEST ROUTINES
                    ;
0093  30    RIGHT:  TSX
0094  BD  00AC'      JSR   SHAPE
0097  DF  00G       STX   R10,D   ;SAVE DATA ADDR
0099  DE  00G       LDX   ZX,D
009B  DF  00G       STX   R11,D   ;I HAVE REAL MATRIX
009D  39           RTS
                    ;
009E  30    LEFT:   TSX
009F  BD  0000G     JSR   A9X     ;SKIP ONE OPERAND ON STACK
00A2  BD  00AC'     JSR   SHAPE
00A5  DF  00G       STX   R8,D
00A7  DE  00G       LDX   ZX,D
00A9  DF  00G       STX   R9,D
00AB  39           RTS
                    ;
00AC  DF  00G   SHAPE:  STX   R2,D    ;SAVE FOR LATER
00AE  EE  07        LDX   7.,X
00B0  9C  00G       CPX   R0,D    ;IS SHAPE OKAY
00B2  26  12        BNE   FLSHAP
00B4  DE  00G       LDX   R2,D
00B6  EE  09        LDX   9.,X
00B8  9C  00G       CPX   R1,D
00BA  26  0A        BNE   FLSHAP
00BC  DE  00G       LDX   R2,D    ;GET STACK ENTRY ADDR
00BE  EE  03        LDX   3,X     ;GET DATA ADDR
00C0  EE  GB        LDX   NTAPTR,X
00C2  BD  0000G     JSR   A5X
00C5  39           RTS
                    ;
00C6  32    FLSHAP: PUL A         ;STRIP OFF OLD RETURN POINTS
00C7  32            PUL A
00C8  32            PUL A
00C9  32            PUL A
00CA  BD  0000G     JSR   SETERA
00CD  00G           .BYTE ERSHAP
```

```
                                        .SBTTL  *** LTIREL      STRING COMPARE (NONE FUZZIE)
                                        .GLOBL  LTIREL
                                ;
                                ;       INPUTS
                                ;       ARGUMENTS ARE ON STACK AS POINTER TO STRING COUNT
                                ;
                                ;       OUTPUTS
                                ;       ONE OR ZERO ON STACK
                                ;
                                ;       NOTES.
                                ;       CIKN HAS THE TOKEN CODE FOR CALLER
                                ;       THIS ROUTINE DEPENDS ONTHE CODE SEQUANCE OF THESE CODES
                                ;
                                ;       R0 - LEFT ARG DATA ADDR-2
                                ;       R1 - LEFT ARG DATA END ADDR-2
                                ;       R2 - RIGHT ARG DATA ADDR-2
                                ;       R3 - LENGTH OF ARGUMENT SWITCH
                                ;               1 - LEFT ARG IS LONGER
                                ;               0 - THEY ARE EQUAL IN LENGTH
                                ;              -1 - RIGHT IS LONGER
                                ;
00CE    30                      LITREL: TSX                     ;GET LEFT ARGUMENT POINTER
00CF    EE      03                      LDX     3,X
00D1    DF      00G                     STX     R0,D
00D3    A6      00                      LDA A   0,X             ;GET LENGTH OF STRING
00D5    E6      01                      LDA B   1,X
00D7    30                              TSX                     ;NOW GET RIGHT ARG JUNK
00D8    EE      08                      LDX     8.,X
00DA    DF      00G                     STX     R2,D
00DC    7F      00000G                  CLR     R3              ;PRESET R3 FOR LEFT=RIGHT
00DF    A1      00                      CMP A   0,X             ;LEFT COUNT TO RIGHT COUNT
00E1    22      0D                      BHI     LRLLNG          ;I NEED SHORTEST LENGTH
00E3    25      06                      BCS     LRRLNG
00E5    E1      01                      CMP B   1,X
00E7    22      07                      BHI     LRLLNG
00E9    27      0C                      BEQ     LRCOM           ;IF EQUAL I AM READY TO GO
00EB    7C      00000G          LRRLNG: INC     R3              ;R3 = 1
00EE    20      07                      BRA     LRCOM
00F0    A6      00              LRLLNG: LDA A   0,X             ;GET SHORT LEN
00F2    E6      01                      LDA B   1,X
00F4    7A      00000G                  DEC     R3              ;R3 = -1
00F7    DB      01G             LRCOM:  ADD B   R0+1,D          ;CALC STOP ADDR FOR LEFT ARG
00F9    99      00G                     ADC A   R0,D
00FB    97      00G                     STA A   R1,D
00FD    D7      01G                     STA B   R1+1,D
                                ;
00FF    30                              TSX                     ;PRUNE TAG
0100    BD      00000G                  JSR     A10X
0103    35                              TXS
                                ;
0104    DE      00G             LRLOOP: LDX     R0,D            ;GET NEXT BYTE ADDR
0106    9C      00G                     CPX     R1,D            ;AM I DONE
0108    27      1A                      BEQ     LRSTOP          ;FIELDS ARE EQUAL FOR LENGTH
010A    A6      02                      LDA A   2,X
010C    08                              INX                     ;UPDATE POINTER
010D    DF      00G                     STX     R0,D
010F    BD      00000G                  JSR     TESTCS
```

*** LTIREL STRING COMPARE (NONE FUZZIE)

```
                0112  16                        TAB
                0113  DE    00G                 LDX     R2,D
                0115  A6    02                  LDA A   2,X
                0117  08                        INX
                0118  DF    00G                 STX     R2,D
                011A  BD    0000G               JSR     TESTCS
                011D  11                        CBA
                011E  22    0E                  BHI     LRGT            ;STOP IF NOT EQUAL
                0120  25    10                  BCS     LRLT
                0122  20    EU                  BRA     LRLOOP
                                    ;
                0124  96    00G     LRSTOP:     LDA A   R3,D            ;IF CHARS ARE EQUAL LENGTH DECIDE
                0126  2E    06                  BGT     LRGT            ;R3 WAS FLAG
                0128  20    08                  BLT     LRLT
                012A  86    01                  LDA A   1,I             ;SET CODE FOR LEFT=RIGHT
                012C  20    06                  BRA     LRCMP
                012E  86    02      LRGT:       LDA A   2,I             ;CODE FOR GT
                0130  20    02                  BRA     LRCMP
                0132  86    04      LRLT:       LDA A   4,I
                0134  97    00G     LRCMP:      STA A   R4,D            ;SAVE CODE FOR LATER
                0136  96    00G                 LDA A   CTKN,D          ;GET CALLING TOKEN
                0138  80    00G                 SUB A   EQUCOD,I        ;MAKE IT 0 TO 5 FOR TABLE LOOK UP
                013A  CE    0150'               LDX     LRTBL,I
                013D  BD    0000G               JSR     LDAX            ;FANCY LOAD A INDEXED
                0140  CE    0000G               LDX     FPONE,I         ;SET UP FOR TRUE TEST
                0143  95    00G                 BIT A   R4,D            ;TRY MASK AGAINST RESULT
                0145  26    03                  BNE     LRTRUE          ;IT WAS TRUE
                0147  CE    0000G               LDX     FPZERO,I        ;SET FALSE
                014A  BD    0000G   LRTRUE:     JSR     PSHFPN          ;PUT IT ON STACK
                014D  7E    0000G               JMP     DREXTA          ;SOO LONG
                                    ;
                0150  01    06      04  LRTBL:    BYTE    1,6,4,5,3,2     ;ORDER MUST MATCH CODES FOR TOKENS
          0153  05    03    02
                                    ;               ------------------------------
```

```
                                        .SBTTL  XXX SUM         PLUS REDUCE FUNCTION
                                        .GLOBL  SUM
                                ;
                                ;       INPUTS
                                ;       ONE MATRIX IS ON THE STACK
                                ;
                                ;       OUTPUTS
                                ;       ONE SCALER VALUE ON THE STACK
                                ;
0156    30              SUM:    TSX                     ;GET NT PTR
0157    EE      01              LDX     1,X
0159    EE      08              LDX     NTAPTR,X        ;GET ADDR OF DATA OBJECT
015B    BD      0000G           JSR     A5X             ;DATA ORIGIN ADDR
015E    DF      00G             STX     R0,D
0160    30                      TSX
0161    EE      03              LDX     3,X             ;END ADDR
0163    BD      0000G           JSR     A5X
0166    DF      00G             STX     R1,D
0168    30                      TSX                     ;PRUNE STACK
0169    BD      0000G           JSR     A9X
016C    35                      TXS
016D    CE      0000G           LDX     #PZERO,I        ;PRIME STACK
0170    BD      0000G           JSR     PSHFPN
0173    DE      00G             LDX     R0,D            ;GET ONE VALUE
0175    BD      0000G   SUMLP:  JSR     PSHFPN
0178    BD      0000G           JSR     FPADD           ;ADD THEM UP
017B    DE      00G             LDX     R0,D
017D    BD      0000G           JSR     A8X             ;NEXT ITME ADDR
0180    DF      00G             STX     R0,D            ;SAVE IT
0182    9C      00G             CPX     R1,D            ;ALL DONE
0184    26      EF              BNE     SUMLP
0186    7E      0000G           JMP     DREXTA          ;ALL DONE
                                ;
                0001'           .END
```

MATOPR SCALER OVER MATRIX OPER RT-11 MMAC VM02-10 14-OCT-76 01:36:07 PAGE 9+
SYMBOL TABLE

```
ABRFLG=  0040        AFAIL =  0030        AFPITT= xxxxxx G     ALLOK =  0004        ALLTG = xxxxxx G
AMAT  =  0010        AREAY =  0020        ASSCOD= xxxxxx G     ASSCSC= xxxxxx G     ASTR  =  0020
ATSNTG= xxxxxx G     A10X  = xxxxxx G     A11X  = xxxxxx G     A12X  = xxxxxx G     A13X  = xxxxxx G
A5X   = xxxxxx G     A6X   = xxxxxx G     A8X   = xxxxxx G     A9X   = xxxxxx G     BAKSTG= xxxxxx G
BMAT  =  0004        BSTR  =  0008        CALLTG= xxxxxx G     COOPTR= xxxxxx G     CDSPTR= xxxxxx G
CLPTR = xxxxxx G     CMAT  =  0001        CONCOD= xxxxxx G     CRCOD = xxxxxx G     CSTR  =  0002
CTKN  = xxxxxx G     DATCOD= xxxxxx G     DIMFLG=  0004        DISSRQ=  0008        DREXTR= xxxxxx G
DREXTB= xxxxxx G     EOFCOD= xxxxxx G     EOFTBL= xxxxxx G     EOLTG = xxxxxx G     EOSTG = xxxxxx G
EQUCOD= xxxxxx G     ERRCGN= xxxxxx G     ERBRK = xxxxxx G     ERDOMN= xxxxxx G     EREOFN= xxxxxx G
ERFORA= xxxxxx G     ERLNNF= xxxxxx G     ERNOT = xxxxxx G     ERNIMX= xxxxxx G     ERNOFN= xxxxxx G
ERNXEN= xxxxxx G     ERNXTA= xxxxxx G     EROFA = xxxxxx G     ERRCD = xxxxxx G     ERRCOB= xxxxxx G
ERSHAP= xxxxxx G     ERSTOP= xxxxxx G     ERUNOF= xxxxxx G     ERVAL = xxxxxx G     ERWSFL= xxxxxx G
ESTG  = xxxxxx G     EXTFLG=  0080        FLSHAP   00C6R       FNACOD= xxxxxx G     FNFLG =  0010
FNTBL = xxxxxx G     FORTG = xxxxxx G     FPROD = xxxxxx G     FPRITT= xxxxxx G     FPC   = xxxxxx G
FPONE = xxxxxx G     FPZERO= xxxxxx G     GLBFLG= xxxxxx G     GOSTG = xxxxxx G     HALTA = xxxxxx G
IMACOD= xxxxxx G     IMXFLG=  0020        IMXTG = xxxxxx G     ITM1TG= xxxxxx G     ITM2TG= xxxxxx G
IMPAX = xxxxxx G     IMPX  = xxxxxx G     KEYFLG=  0010        KEYSTK= xxxxxx G     LBRKTG= xxxxxx G
LCLFLG= xxxxxx G     LDAX  = xxxxxx G     LDBX  = xxxxxx G     LDXX  = xxxxxx G     LEFT     009EP
LISTTG= xxxxxx G     LITCOD= xxxxxx G     LITREL   00CERG      LNNOTG= xxxxxx G     LOOP     0038R
LRCMP    0134R       LRCOM    00E7R       LT2GT    012ER       LRLLNG   00E0R       LRLOOP   0104R
LRLT     0132R       LRRLNG   00E8R       LRSTOP   0124R       LRTBL    0150R       LRTRUE   014AR
LSP   = xxxxxx G     LSTCOD= xxxxxx G     MATMAT   0011RG      MATOPR=  0000RG      MATSCL   001CRG
MMICOD= xxxxxx G     MPLCOD= 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=  0006        NULLTG= xxxxxx G     OBJATR=  0002        OBJBCK=  0003
OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002        ONTBL = xxxxxx G     OPRADR= xxxxxx G
OPRCL = xxxxxx G     OUAFLG=  0002        OUTONE   0072R       PRETG = xxxxxx G     PARM  =  0008
PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMEP =  0003        PGMLEN=  0000
PGMLNN=  0007        PGMPTR= xxxxxx G     PGMTG = xxxxxx G     PLOSTG= xxxxxx G     PNDEOF= xxxxxx G
PNDFLG= xxxxxx G     PNTNTG= xxxxxx G     PNTSTG= xxxxxx G     PRTTG = xxxxxx G     PSCTG = xxxxxx G
PSHEPN= xxxxxx G     PULEPN= xxxxxx G     RESULT   006ER       REAIL =  0040        RIGHT    009BR
RMAT  =  0040        RSTR  =  0080        RTRNTG= xxxxxx G     RUNFLG=  0080        R0    = xxxxxx G
R1    = xxxxxx G     R10   = xxxxxx G     R11   = 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     SP    = xxxxxx G     SCALER=  0040        SCLMAT   002ARG      SENTTG= xxxxxx G
SETERA= xxxxxx G     SETERR= xxxxxx G     SHAPE    00ACR       SIZCOD= xxxxxx G     SKIPA    004AR
SKIPB    005AR       SNGMAT   0000RG      STAX  = xxxxxx G     STPFLG=  0040        STPTR = xxxxxx G
STRING=  0010        SUM      0156RG      SUMLP    0175R       TESTCS= xxxxxx G     TRCFLG=  0020
UNDEF =  0080        VALERR=  0040        VALTG = xxxxxx G     ZX    = xxxxxx G
  ABS.   0000   00
         0189   01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2595. WORDS
.SY:MATOPR/C(DK1:SEICLI,MATOPR
```

CROSS REFERENCE TABLE (CREF V01-03.)

	2-5					
A10X	7-8#	7-66	8-48			
A11X	7-8#					
A12X	7-8#					
A13X	7-8#					
A5X	7-8#	7-110	7-114	7-152	-13	9-17
A6X	7-8#					
A8X	7-8#	7-75	7-82	7-90	9-28	
A9X	7-8#	7-106	7-134	9-20		
ABRFLG	2-10#					
AFAIL	5-38#					
AFPITT	7-11#					
ALLOK	4-17#					
ALLTG	5-24#					
AMAT	5-37#					
ARRAY	4-10#					
ASGCOD	1-36#					
ASSCSC	7-9#					
ASTR	5-36#					
ATSNTG	5-22#					
BAKSTG	5-23#					
BMAT	5-40#					
BSTR	5-39#					
CALLTG	5-27#					
CDOPTR	3-40#					
COSPTR	3-39#					
CLPTR	2-4#					
CMAT	5-42#					
CONCOD	1-45#					
CRCOD	1-44#					
CSTR	5-41#					
CTRN	2-7#	8-78				
DATCOD	1-47#					
DIMFLG	2-13#					
DISSRQ	2-21#					
DREXTA	2-25#	8-87	9-32			
DREXTB	2-26#					
EOFCOD	1-41#					
EOFTBL	3-5#					
EOLTG	5-28#					
EOSTG	5-29#					
EQUCOD	1-37#	8-79				
ERASGN	6-16#					
ERBRK	6-14#					
ERDOMN	6-7#					
EREOFN	6-15#					
ERFORA	6-11#					
ERLNNF	6-8#					
ERNOT	6-13#					
ERNTMX	6-5#					
ERNOFN	6-4#					
ERNXFN	6-20#					
ERNXTA	6-12#					
EROFA	6-10#					
ERRCD	2-31#					
ERRCDB	2-32#					

CROSS REFERENCE TABLE (CREF V01-03)

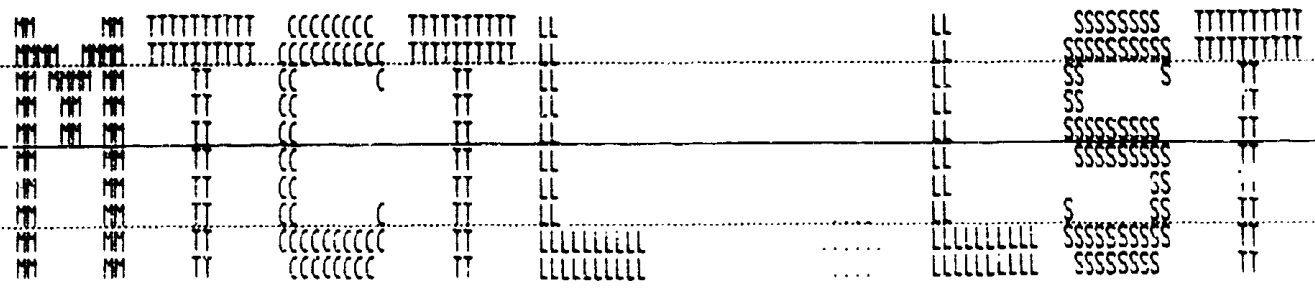
CROSS REFERENCE TABLE (CREF V01-03)

MULCOD	1-40#											
NCSFLG	2-23#											
NLPTR	2-5#											
NTAPTR	4-20#	7-109	7-151	9-12								
NTATTR	4-7#											
NTDIMS	4-19#											
NTDLEN	4-22#											
NTLINK	4-4#											
NTNAME	4-5#											
NTPTR	2-6#											
NTSPTR	4-24#											
NTVAL	4-15#											
NTWCOL	4-18#											
NTWLEN	4-23#											
NTWROW	4-17#											
NULLTG	5-4#											
OBJATR	4-30#											
OBJBCK	4-31#											
OBJDT	4-32#											
OBJLEN	4-28#											
ONSFLG	2-14#											
ONTBL	3-4#											
OPRADR	2-24#											
OPRCL	7-12#	7-87										
OVAFLG	2-22#											
OUTONE	7-37	7-107#										
PAETG	5-15#											
PCRM	4-12#											
PGMATR	4-37#											
PGMBP	4-39#											
PGMCD	4-41#											
PGMFP	4-38#											
PGMLEN	4-36#											
PGMLNN	4-40#											
PGMPTR	2-50#											
PGMTG	5-10#											
PLOSTG	5-5#											
PNDEOF	2-47#											
PNDFLG	2-33#											
PNTNTG	5-14#											
PNTSTG	5-12#											
PRTTG	5-26#											
PSCTG	5-13#											
PSHFPN	7-9#	7-74	7-81	8-86	9-23	9-25						
PULFPN	7-9#	7-89										
R0	2-28#	7-118*	7-143	8-24*	8-42	8-43	8-51	8-56*	9-14*	9-24	9-27	9-29*
R1	2-28#	7-121*	7-147	8-44*	8-45*	8-52	9-18*	9-30				
R10	2-29#	7-57*	7-80	7-83*	7-86*	7-128*						
R11	2-29#	7-58*	7-84	7-130*								
R2	2-28#	7-107*	7-112	7-116	7-119	7-141*	7-145	7-149	8-29*	8-59	8-62*	
R3	2-28#	8-30*	8-37*	8-41*	8-69							
R4	2-28#	8-77*	8-83									
R5	2-28#											
R6	2-28#	7-88	7-91*	7-111*								
R7	2-28#	7-92	7-115*									
R8	2-29#	7-40*	7-67*	7-72	7-76*	7-79*	7-136*					
R9	2-29#	7-41*	7-68*	7-77	7-138*							

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

PLRA 1-20#
PLRB 1-27#
SEI 1-3#



MTCTL MAG TAPE CONTROLLER RT-11 MMAC VM02-10 14-OCT-76 01:36:21
TABLE OF CONTENTS

```
                        .TITLE  MTCTL   MAG. TAPE CONTROLLER
                        .IDENT  /SBN:03/
                        .GLOBL  TAPE            ; MAIN MAG. TAPE CONTROL SECTION
                        .GLOBL  KILL            ; KILL ENTRY FROM EVAL
                        .GLOBL  SPCIOF          ; SPECIAL I/O FUNCTIONS (PRINT LIKE)
                        .GLOBL  TYPARG          ; ARGUMENT TYPE CHECK
                        .GLOBL  OLD             ; "OLD" ENTRY FROM EVAL
                        .GLOBL  BAKSRS          ; MT DRIVER TO BCK UP ONE REC
                        .GLOBL  A14X            ; ONE OF THOSE
                        .GLOBL  SERCHS          ; MT DRIVER TO SEARCH FOR A FILE.
                        .GLOBL  REWIND          ; MT DRIVER TO REWIND TAPE
                        .GLOBL  READRS          ; MT DRIVER TO READ A REC.
                        .GLOBL  WRRS            ; MT DRIVER TO WRITE A REC.
                        .GLOBL  MARKS           ; MT DRIVER TO MARK A TAPE
                        .GLOBL  MTCHKS          ; MT CHECK SUM GENERATOR
                        .GLOBL  ADRDEV          ; ADDRESS THE DEVICE
                        .GLOBL  BERALC          ; ALLOCATE BUFFER
                        .GLOBL  MTPTR           ; MAGTAPE BUFFER POINTER
                        .GLOBL  MTMAX           ; END OF PRESENT MAGTAPE BUFFER
                        .GLOBL  MTCLOS          ; ROUTINE TO CLOSE MAGTAPE
                        .GLOBL  FIXNUM          ; FIXES INTEGER NUMBER
                        .GLOBL  PSHFPN
                        .GLOBL  PULFPN          ; GOOD OLD FLOATING POINT ROUTINES
                        .GLOBL  FIX1
                        .GLOBL  A9X
                        .GLOBL  INBITT          ; INTEGER -) ASCII (INTERNAL)
                        .GLOBL  AFPITT          ; ASCII -) FPN (INTERNAL)
                        .GLOBL  MTNULL          ; NULL MAGTAPE BUFFER
                        .GLOBL  FILEND          ; FILE TO BE FOUND
                        .GLOBL  FILLOC          ; PRESENT FILE
                        .GLOBL  OPRADR          ; KEY WORD FROM EVAL
                        .GLOBL  MTBER           ; MAGTAPE BUFFER
                        .GLOBL  KILTYP          ; KILL SEED CONSTANT
                        .GLOBL  MTMARK          ; MAG. TAPE MARK DRIVER
                        .GLOBL  MTKILL          ; MAG TAPE KILL DRIVER
                        .GLOBL  MTFIND          ; MAG TAPE FIND DRIVER
]]]]]
                        .GLOBL  EDTPTR          ; EDIT POINTER (CURSOR)
                        .GLOBL  MOMTH           ; MAKES MAGTAPE HEADERS
                        .GLOBL  FPC             ; TEMP FLOATING REG.
                        .GLOBL  SNDBFR          ; SEND BUFFER
                        .GLOBL  MAGEND          ; HARDWARE FINISHED FLAG
                        .GLOBL  ERMTRD          ; MAGTAPE READ ERROR
                        .GLOBL  ERMFMT          ; MAGTAPE FORMAT ERROR
                        .GLOBL  ERMOPN          ; ILLEGAL MAGTAPE REQUEST WILL TAPE OPEN
                        .GLOBL  ERFNFD          ; MAGTAPE FILE NOT FOUND
                        .GLOBL  ERMILA          ; MAG TAPE ILLEGAL ARGUMENT
                                                ; OR ILLEGAL ACCESS
                ;*****************************************************************
                ;       SOME CONSTANTS
                ;
                        .GLOBL  MTSTAT          ; MAG TAPE STATUS
                        .GLOBL  MTFSIZ,MTFPOS,MTSNEW,MTSASC,MTSBIN,MTSPGM,MTSSEC
                        .GLOBL  MTOPEN
0080                    MTFSIZ  = H80           ; PRESENT REC. SIZE     0 = 256 BYTE
                                                ;                       1 = 128 BYTE
0040                    MTFPOS  = H40           ; ACCESS OF FILE        1 = FIRST ACCESS TO FILE
                                                ;                       0 = .NOT. 1
```

```
0020        MTOPEN = H20          ; SET IF MAGTAPE FILE IS OPEN
0010        MTSNEW = H10          ; NEW FILE IS PRESENT IF SET
0008        MTSASC = H08          ; ASCII FILE PRESENT
0004        MTSBIN = H04          ; BINARY FILE PRESENT
001C        MTSLST = H1C          ; IF ABOVE 3 ARE CLEAR THEN FILE IS LAST
0002        MTSPGM = H02          ; IF SET THEN IS PROGRAM FILE OTHERS
                                  ; ARE DATA FILES
0001        MTSSEC = H01          ; SET IF PRESENT FILE IS SECRET


            .GLOBL MTSTT2         ; MAG TAPE STATUS BYTE 2
            .GLOBL MTFFST,MTSRD,MTSWRT
            .GLOBL YAXIS          ; USING THIS AS STATUS BYTES
0080        MTFFST = H80          ; FIRST RECORD ON FILE IS UNDER HEADS
                                  ; OR HEAD BETWEEN FILES
0010        MTSRD  = H10          ; PRESENT MAGTAPE BUFFER VALID FOR READ
0002        MTSWRT = H02          ; PRESENT FILE OPENED AND WRITTEN INTO

            .GLOBL MTSREG
0001        MTFLEN = 1
0002        MTCKSM = 2
0004        MTNHDR = 4
```

```
                                        .SBTTL  TAPE----GENERAL TAPE COMMAND DISPATCH
                                ; MARK AND FIND ARE GENERAL IEC COMMANDS
                                ; AND CAN BE MADE TO LOOK LIKE A PRINT
                                ;FIND, MARK, TLIST(?) CALL THIS ENTRY POINT FROM EVAL
                                        .GLOBL  IOSCAN
                                        .GLOBL  SPCIOB
                                        .GLOBL  IOCLNR
                                ;
0000    86      000     TAPE:   LDA A   EOLTG,I         ; GET A TAG TO FIND LATER
0002    36                      PSH A
0003    9F      00G             STS     R0,D
0005    BD      0000G           JSR     ADRDEV          ; GO ADDRESS AS A MAGTAPE
0008    BD      0000G           JSR     BFRALC
000B    9F      00G             STS     R0,D
000D    86      10              LDA A   16.,I
000F    97      00G             STA A   IOFUNC,D
0011    BD      0000G           JSR     SPCIOB          ; GO DO STACK SCAN
0014    BD      0000G           JSR     UNADR           ; THAT REALLY CLEANS IT UP
0017    BD      0000G           JSR     IOCLNR          ; CLEAN UP STACK AND EXIT
```

```
                                        .SBTTL  KILL----KILL AS CALLED FROM EVAL
                                ;THIS ROUTINE DOES THE PREDISPATCH CHECKING OF THOSE I/O COMMANDS THAT
                                ;HAVE TAPE, IEC, AND FILE DEVICE IN COMMON.
                                ;THOSE ARE: 1) OLD, 2) SAVE, 3) KILL, AND 4) LIST (IN SOME
                                ;SENSE.
                                ;
                                        .GLOBL  TAPFIL
                                        .GLOBL  FILES
                                        .GLOBL  KILTYP          ; KILL SEED
                                        .GLOBL  SETARG          ; SETUP FOR TYPARG CALL + TYPARG
                                ;
001A  CE  0000G   KILL:  LDX    KILTYP,I        ; SET UP KILL SEED
001D  DF    00G          STX    OPRADR,D
001F  30                 TSX
0020  08                 INX
0021  DF    00G          STX    R0,D
0023  BD  0000G          JSR    CLRARG          ; FIND TYPE OF ARGUMENT ON STACK
                                        .GLOBL  CLRARG
0026  4D                 TST A
0027  27    07          BEQ    TAPE
0029  7E  0000G   KIFILE: JMP    FILES
```

```
                                        .SBTTL  APPOLD--APPEND FORM OF OLD
                                    ;APPEND HAS ALLREADY PROCESSED IT'S INFORMATION AND HAS LEFT
                                    ;SOME INFORMATION ON THE STACK (36 BYTES WORTH).
                                    ;
                                        .GLOBL  APPOLD
                                        .GLOBL  TAPF2           ; SECOND ENTRY IN TAPFIL
                                        .GLOBL  OLD
                                        .GLOBL  OPRADR
002C    86      00G     APPOLD: LDA A   EOLTG,I         ; PUSH MY EOL TAG
002E    36                      PSH A
002F    30                      TSX                     ; SKIP APPENDS STUFF
0030    BD      0000G           JSR     A11X
                                .GLOBL  A11X
0033    DF      00G             STX     R0.D            ; SAVE STACK LOCATION
0035    CE      0000G           LDX     OLD,I           ; SET UP I/O PROC. SO IT KNOWS
0038    DF      00G             STX     OPRADR,D        ; WHATS COMMING OFF!
003A    BD      076F'           JSR     TAPF2           ; THIS WILL CREATE A SECOND EOL
                                                        ; IN ADDITION TO THE EVALUATORS
                                                        ; AND CONTROL COMES BACK HERE SO I CAN
                                                        ; CLEAN ONLY MY STUFF OFF THE STACK
003D    31                      INS                     ; REMOVE MY EOL TAG
003E    39                      RTS                     ; RETURN TO APPEND
```

```
                                        .SBTTL  MODMTH--MODIFY MAG TAPE HEADER
                                ; INDEX = POINTER TO NEW HEADER INFO
                                ; A.STRT = POINTER TO BEGINNING OF BUFFER
                                ;       A = LENGTH
                                ;       B = POSITION IN MTBER
                                ;       R0 IS USED FOR TEMP. STORE
                                ;
                                        .GLOBL  MODT88,MODMTH
003F    86      08      MODT88: LDA A   8.,I
0041    16                      TAB                     ; SPECIAL ENTRY POINT BY G.I.R
0042    DB      01G     MODMTH: ADD B   A.STRT+1,D      ; GET POINTER TO PROPER PLACE IN BUFFER
0044    D7      01G             STA B   R1+1,D
0046    D6      00G             LDA B   A.STRT,D
0048    C9      00              ADC B   0,I
004A    D7      00G             STA B   R1,D
004C    E6      00      MODLOP: LDA B   0,X             ; GET CHAR. TO XFER.
004E    08                      INX
004F    DF      00G             STX     R0,D            ; SAVE POINTER INTO TABLE FOR NEW HEADER
0051    DE      00G             LDX     R1,D            ; RECALL HEADER POINTER
0053    E7      00              STA B   0,X             ; STORE CHAR.
0055    08                      INX
0056    DF      00G             STX     R1,D            ; SAVE POINTER
0058    DE      00G             LDX     R0,D            ; RECALL TABLE POINTER
005A    4A                      DEC A                   ; DONE?
005B    26      EF              BNE     MODLOP
005D    39                      RTS                     ; EXIT
                                ;*****************************************************************
                                ;       MAG TAPE HEADER TABLE
                                ;
                                        .GLOBL  TYPNEW,TYPBIN,TYPASC,TYPLST,PRGBLN,PRGPGM,PRGDAT,TYPSEC
005E    4E      45      57 TYPNEW: .ASCII  /NEW     /
0061    20      20      20
0064    20      20
0066    42      49      4E TYPBIN: .ASCII  /BINARY  /
0069    41      52      59
006C    20      20
006E    41      53      43 TYPASC: .ASCII  /ASCII   /
0071    49      49      20
0074    20      20
0076    4C      41      53 TYPLST: .ASCII  /LAST    /      ; MARKS LAST FILE ON TAPE
0079    54      20      20
007C    20      20
007E    20      20      20 PRGBLN: .ASCII  /        /
0081    20      20      20
0084    20      20      20
0087    20
0088    50      52      4F PRGPGM: .ASCII  /PROGRAM   /
0088    47      52      41
008E    4D      20      20
0091    20
0092    44      41      54 PRGDAT: .ASCII  /DATA      /
]]]]]
0095    41      20      20
0098    20      20      20
009B    20
009C    53      45      43 TYPSEC: .ASCII  /SECRET  /
009F    52      45      54
00A2    20      20
```

```
                                        .SBTTL  MTKILL--MAG. TAPE KILL ROUTINE
                                ;THIS ROUTINE IS CALLED FROM THE SPECIAL PRINT PROCESSOR
                                ;AND PREFORMS THE KILL FUNCTION TO THE MAGTAPE
                                ;THIS FUNCTION CONSISTS IN FINDING THE REQUESTED FILE AND REMARKING
                                ;THE HEADER TO MAKE IT A NEW FILE
00A4  BD  021!'  MTKILL: JSR     MTFIND          ; FIND PROPER PLACE
00A7  96    00G         LDA A   ERRCD,D         ; ERROR?
00A9  26    10          BNE     MTKEXX
00AB  96    00G         LDA A   MTSREG,D        ;THIS IS THE LAST CORRECTION
00AD  85    04          BIT A   MTNHDR,I        ;THAT STEVE MADE
00AF  26    0A          BNE     MTKEXX          ;IN SEPTEMBER
00B1  96    00G         LDA A   MTSTAT,D        ;OR SO HE SAID
00B3  85    1C  MTKHDR: BIT A   MTSNEW+MTSASC+MTSBIN,I  ; VALID FILE?
00B5  26    05          BNE     MTKOK
00B7  86    00G         LDA A   ERFNFD,I        ; IF TRYING TO KILL LAST FILE THEN ERROR
00B9  97    00G         STA A   ERRCD,D
00BB  39            MTKEXX: RTS
00BC  BD  0208' MTKOK:  JSR     NFRS            ; BACK UP TO HEADER AGAIN
00BF  CE  005E'         LDX     TYPNEW,I        ; NEW HEADER INFO
00C2  BD  003E'         JSR     MODI88          ; GO MODIFY MAGTAPE HEADER
00C5  CE  007E'         LDX     PRGBLN,I        ; MAKE PROGRAM TYPE BLANK
00C8  86    0A          LDA A   10.,I
00CA  C6    10          LDA B   16.,I
00CC  BD  0042'         JSR     MODMTH
00CF  CE  007E'         LDX     PRGBLN,I        ; SECRET FIELD BLANKED
00D2  86    08          LDA A   8.,I
00D4  C6    1A          LDA B   26.,I
00D6  BD  0042'         JSR     MODMTH
00D9  BD  0393'         JSR     MTWRT           ; WRITE IT
00DC  7F  00DCG         CLR     MTSTAT
00DF  39                RTS
```

```
                                        .SBTTL  MTMARK--MAG. TAPE MARK ROUTINE
                                        .GLOBL  MTXBYT
                                        .GLOBL  INT,ABS
                                ;THIS ROUTINE IS CALLED FROM THE SPECIAL PRINT PROCESSOR
                                ;AND IS NORMALLY CALLED TWICE.  THE FIRST CALL SETS NUMBER OF
                                ;FILES TO MARK.  THE SECOND CALL SETS LENGTH OF EACH FILE
                                ;AND PROCEEDS TO MARK THE FILES.  IF IN NORMAL MODE EACH FILE
                                ;WILL HAVE A HEADER (MARKED AS NEW) AND WILL CONSIST OF RECORDS
                                ;ALSO BE WRITTEN WITH EACH RECORD.  IF  MODE IS IN EFFECT NO HEADER
                                ;WILL BE WRITTEN AND EACH RECORD WITHIN THE FILE WILL BE 128 BYTES
                                ;PLUS A 2 BYTE CRC CHECK.
                                ;
            000D                        CR      = %00
            0013                        DC3     = %3
                                ;
00E0  96  000       MTMARK: LDA A   YAXIS,D         ; GET MAGTAPE STATUS
00E2  26  13                BNE     MRKIT           ; GO AND DO IT
00E4  43                    COM A
00E5  97  000               STA A   YAXIS,D
00E7  BD  0000G             JSR     FIXNUM          ; FIX THE NUMBER
00EA  29  06                BVS     MTM3
00EC  25  04                BCS     MTM3
00EE  DF  000               STX     R8,D            ; SAVE # OF FILES
00F0  26  04                BNE     MTM1            ; IF # OF FILES = 0 THEN ERROR
00F2  C6  000       MTM3:   LDA B   ERDOMN,I
00F4  D7  000       MTM2:   STA B   ERRCD,D
00F6  39            MTM1:   RTS
                            ;
                            ;INFORMATION:
                            ;       R8+1 = # OF FILES REQUESTED
                            ;       R9 = # OF RECORDS PER FILES
                            ;
00F7  B6  0000G     MRKIT:  LDA A   PIAMTA          ; TEST IF WRITE LOCKED
00FA  85  10                BIT A   20,I
00FC  27  04                BEQ     1$
00FE  C6  000               LDA B   ERWRT,I
0100  20  F2                BRA     MTM2
0102  96  000       1$:     LDA A   MTSTAT,D        ; IF FILE OPEN THEN ILLEGAL ACCSESS
0104  85  40                BIT A   MTFPOS,I
0106  27  04                BEQ     MTMOK1
0108  C6  000               LDA B   ERMOPN,I ;XXXXXX ADDED ,I - BOB
010A  20  E8                BRA     MTM2
010C  DE  000       MTMOK1: LDX     POINT,D         ; GET FLOATING POINT NUMBER OF BYTES
010E  BD  0000G             JSR     PSHFPN
0111  30                    TSX
0112  C6  F9                LDA B   -7,I
0114  96  000               LDA A   MTSREG,D
0116  85  01                BIT A   MTFLEN,I
0118  26  01                BNE     1$
011A  5A                    DEC A
011B  EB  02        1$:     ADD B   2,X
011D  E7  02                STA B   2,X
011F  A6  01                LDA A   1,X
0121  8A  80                ORA A   128.,I
0123  25  01                BCS     MRFIX
0125  4A                    DEC A
0126  A7  01        MRFIX:  STA A   1,X
```

```
                                JSR     INT
                                JSR     ABS
                                TSX
                                JSR     FIX1
                                LDA A   FIX1A,X
                                LDA B   FIX1B,X
                                LDX     FIX1A,X         ; GET # OF RECORDS
                                TST A
                                BNE     1$
                                SUB B   3,I
                                BCC     1$
                        2$:     INX
                                INC B
                                BNE     2$
                        1$:     STX     R9,D            ; SAVE THIS INFO
                                TSX                     ; CLEAN UP STACK
                                JSR     A9X
                                TXS
                                .GLOBL  MKFILE          ; FOR ROM PAKS
                        MKFILE: LDA A   ERRCD,D
                                BNE     MTM3
                                LDA A   FILLOC+1,D      ; IF ON FILE ZERO THEN SET IN FILE GAP
                                BNE     3$
                                LDA B   #TFFST,I
                                STA B   MTSTT2,D
                                INC     FILLOC+1
                        3$:     LDA B   MTSTT2,D        ; CHECK POSITION
                                BMI     MKFIL.
                                JSR     #ER8            ; IF NOT IN GAP PUT IT THERE
                        MKFIL.: LDA A   MTSREG,D        ;   MODE = NO HEADER
                                BIT A   MTNHDR,I
                                BEQ     MKHDR
                                CLR A
                                JSR     MTNULL          ; NULL FILL THE RECORD
                                BRA     MKFIL
                        ;*******************************************************
                        ;       A NEW FILE HEADER
                        ;
                        MKHDR:  LDA A   #<' >),I        ; GET A SPACE TO FILL WITH
                                JSR     MTNULL          ; FILL THAT BUFFER
                                LDX     #TYPNEW,I       ; MAKE UP NEW HEADER
                                JSR     MODT88          ; MAKES THE HEADER IN THE BUFFER
                                LDX     A.STRT,D        ; SET IN THE FILE NUMBER
                                STX     R1,D
                                LDX     #8.,I           ; GET MAX OUTPUT LENGTH
                                STX     R2,D
                                LDX     FILLOC,D        ; GET PRESENT FILE NUMBER
                                STX     R0,D
                                JSR     INAITT          ; INTEGER -> ASCII (INTERNAL XFER)
                                LDX     #MTBFR+34.,I    ; ADD MARKER LENGTH INFO
                                STX     R1,D
                                LDX     #8.,I
                                STX     R2,D
                                LDX     R9,D
                                STX     R0,D
                                JSR     INAITT
                                LDX     #MTBFR+42.,I    ; "CR"
```

```
0198   86   00                LDA A   CR,I
019D   A7   00                STA A   0,X
019F   86   13                LDA A   DC3,I         ; "DC3"
01A1   A7   01                STA A   1,X
01A3   86   FF       MRKFIL:  LDA A   377,I         ; TELL'S HARDWARE THIS ISN'T LAST FILE
01A5   D6   01G               LDA B   R8+1,D
01A7   26   0D                BNE     MRKITT        ; IF NOT LAST ONE THEN JUST MARK IT
01A9   4F                     CLR A                 ; NOW IT'S LAST FILE
01AA   D6   00G               LDA B   MTSREG,D
01AC   C5   04                BIT B   MTNHDR,I
01AE   26   06                BNE     MRKITT
01B0   CE   0076'             LDX     TYPLST,I      ; MAKE IT LAST FILE
01B3   8D   003F'             JSR     MODT88        ; COMPLETE WITH HEADER
01B6   97   00G      MRKITT:  STA A   R10,D
01B8   DE   00G               LDX     A.STRT,D      ; SET UP RECORD FOR MARKS
01BA   DF   00G               STX     MTPTR,D
01BC   86   1A                LDA A   26.,I         ; 10. FOR FUDGE
01BE   97   00G               STA A   MTXBYT,D
01C0   8D   0461'             JSR     MTCHKS        ; FORM CHECKSUM
01C3   96   00G               LDA A   MTSREG,D      ; SSSEE IF NEED A CHECK SUM
01C5   85   02                BIT A   MTCKSM,I
01C7   26   03                BNE     2$
01C9   E7   01                STA B   1,X
01CB   08                     INX
01CC   DF   00G      2$:      STX     MTMAX,D
01CE   DE   00G               LDX     R9,D          ; GET # OF RECORDS
01D0   96   00G               LDA A   MTSREG,D      ; NO HEADER?
01D2   85   04                BIT A   MTNHDR,I
01D4   26   01                BNE     6$
01D6   08                     INX                   ; ADD ONE TO MAKE ROOM FOR HEADER INFO
01D7   DF   00G      6$:      STX     FILFND,D
01D9   BD   0000G             JSR     MARKS         ; MARK IT
01DC   96   00G               LDA A   ERRCD,D       ; OVER RUN ERRORS?
01DE   26   09                BNE     1$
01E0   7C   0001G             INC     FILLOC+1      ; BUMP FILE LOCATION COUNTER
01E3   26   07                BNE     9$
01E5   86   00G               LDA A   ERDOMN,I      ; MAG TAPE ARG. ERRROR
01E7   97   00G               STA A   ERRCD,D
01E9   7E   0000G    1$:      JMP     REWIND        ; REWIND IT, AND RETURN.
01EC   96   01G      9$:      LDA A   R8+1,D        ; NUMBER OF FILES
01EE   27   0D                BEQ     MRKDNN        ; DONE?
01F0   4A                     DEC A
01F1   97   01G               STA A   R8+1,D        ; SAVE FILE LOCATION COUNTER
01F3   26   05                BNE     MRKOK         ; IF LAST ONE THEN IT IS ZERO LENGTH
01F5   CE   0003              LDX     3,I           ; MAKE IT THREE LONG
01F8   DF   00G               STX     R9,D
01FA   7E   0160'    MRKOK:   JMP     MKFIL         ; GO MARK ANOTHER
01FD   7A   0001G    MRKDNN:  DEC     FILLOC+1      ; BACK UP TO LAST FILE
0200   8D   06                BSR     NFR8
0202   BD   0358'             JSR     MTCLOS
0205   7E   0000G             JMP     PUPTAP
```

```
                                        .SBTTL  MTFIND--MAGTAPE FIND ROUINTE
                                ;THIS ROUINTE IS CALLED FROM THE SPECIAL PRINT FUNCTIONS
                                ;AFTER A FIND HAS BEEN REQUESTED.  A FIND PREFORMS THE FOLLOWING:
                                ;       1) LOCATE THE REQUESTED FILE IF IT EXISTST
                                ;       2) READ THE HEADER IF ONE EXISTS
                                ;       3) SET UP MTSTAT DEPENDING ON THE CONTENT OF THE HEADER (OR NOT
                                ;          HEADER)
                                ;
                                ;NOTE:  IT ALSO CLOSES ANY PENDING MT FILE
                                ;
                                        .GLOBL  MTAFIN          ; FOR EXTERNAL ROM PACKS
                                        .GLOBL  BPIAMTB         ; MAGTAPE PIA BUFFER
                                        .GLOBL  BPIAMTB
                                        .GLOBL  TST8NF
                                        .GLOBL  PIAMTA
                                        .GLOBL  RECCNT          ; NUMBER OF AVAILABLE RECORDS ON
                                                                ;       PRESENT FILE.
                                ;
                                ;*****************************************************************
                                ;       SLOW SEARCH FOR FILE MARK
                                ;
0208  BD  0000G   NFR8:  JSR     BAKARS
0208  BD  0000G          JSR     TST8NF          ; LOOK TO SEE IF 8NFRS
020E  20  F8             BRA     NFR8
0210  39                 RTS
                                ;*****************************************************************
                                ;       MAIN FIND ENTRY POINT
                                ;
0211  BD  0000G   MTFIND: JSR     FIXNUM          ; GET NUMBER
0214  29  02              BVS     1$
0216  24  05              BCC     MTAFIN
0218  86  00G     1$:     LDA A   ERDOMN,I        ; DOMAIN ERROR
021A  97  00G             STA A   ERRCD,D
021C  39                  RTS
021D  D7  01G     MTAFIN: STA B   TABPTR+1,D       ; TEMP STORE FIND NUM
021F  BD  0358'           JSR     MTCLOS          ; CLOSE OPEN FILES
0222  96  01G             LDA A   TABPTR+1,D      ; RETRIEVE FILE TO FIND AGAIN
0224  97  01G             STA A   FILFND+1,D      ; SET INTO FILE TO BE FOUND
0226  D6  01G             LDA B   FILLOC+1,D      ; GET PRESENT FILE LOCATION
0228  27  19              BEQ     MTA0
022A  81  01              CMP A   1,I             ; IF NEED TO FIN' 0 OR 1 THEN SPECIAL
022C  27  03              BEQ     RWNDIT
022E  4D                  TST A
]]]]]
022F  26  19              BNE     MTFSRC
0231  36                  PSH A                   ; SAVE LOCATION
0232  BD  0000G           JSR     REWIND
0235  32                  PUL A
0236  D6  00G             LDA B   ERRCD,D
0238  26  04              BNE     1$
023A  97  01G             STA A   FILLOC+1,D      ; SAVE LOCATION
023C  26  13              BNE     MTFRD           ; IF LOOKING FOR #1 THEN PROCEED
023E  39          1$:     RTS                     ; ELSE EXIT
023F  BD  00?JG   MTA0XX: JSR     REWIND
0242  39          MTA0X:  RTS                     ; IF ZERO THEN EXIT
0243  4D          MTA0:   TST A                   ; IF LOOKING FOR 0
0244  27  FC              BEQ     MTA0X           ; ALREADY THERE
0246  81  01              CMP A   1,I             ; IF FILE 0 OR 1 THEN SPECAIL
0248  27  07              BEQ     MTFRD           ; READ IF LOOKING FOR ONE
```

```
0248    BD  0000G   MTFSRC: JSR    SERCHS        ; POSITION TAPE
024D    96  00G             LDA A  ERRCD,D       ; ERRORS?
024F    26  F1              BNE    MTAOX
0251    D6  00G     MTFRD:  LDA B  MTSREG,D      ; IS THERE ANY HEADER?
0253    C5  04              BIT B  MTNHDR,I
0255    27  11              BEQ    MT.256
0257    CE  FFFF            LDX    65535.,!      ; SET FILE LENGTH
025A    DF  00G             STX    RECCNT,D
025C    96  01G             LDA A  FILFND+1,D    ; SSSET FILFND=>FILLOC
025E    97  01G             STA A  FILLOC+1,D
0260    86  80              LDA A  MTFFST,I
0262    97  00G             STA A  MTSTT2,D      ; SSSSET IN GAP BIT
0264    86  08              LDA A  MTSASC,I      ; SET ASCII DATA
0266    20  70              BRA    MTFDON        ; AND ASSUME PROPER FILE
0268    BD  02E8'   MT.256: JSR    MTREAD        ; READ IT
026B    96  00G             LDA A  ERRCD,D       ; ERRORS?
026D    26  D3              BNE    MTAOX
026F    DE  00G             LDX    A.STRT,D      ; SET UP FOR CONVERSION
0271    DF  00G             STX    R0,D
0273    DE  00G             LDX    A.MAX,D
0275    DF  00G             STX    R1,D
0277    BD  0000G           JSR    AFPITT        ; ASCII->FPN INTERNAL
027A    CE  FFFFG           LDX    FPC-1,I
027D    BD  0000G           JSR    FIX1          ; FIX IT
0280    27  06              BEQ    MT.1
0282    86  00G     MT.ERF: LDA A  ERMFMT,I      ; SET IT TO MAG TAPE FORMAT ERROR
0284    97  00G             STA A  ERRCD,D
0286    20  B7              BRA    MTAOXX
0288    E6  04      MT.1:   LDA B  FIX1B,X       ; GET LOCATION FOUND
028A    D1  01G             CMP B  FILFND+1,D    ; IS IT THE ONE WE ARE LOOKING FOR?
028C    26  F4              BNE    MT.ERF
028E    D7  01G     FNDOK:  STA B  FILLOC+1,D
0290    DE  00G             LDX    A.STRT,D      ; CONVERT NUMBER OF RECORDS IN FILE
0292    BD  0000G           JSR    A14X
0295    DF  00G             STX    R0,D
0297    DE  00G             LDX    A.MAX,D
0299    DF  00G             STX    R1,D
029B    BD  0000G           JSR    AFPITT
029E    CE  FFFFG           LDX    FPC-1,I
02A1    BD  0000G           JSR    FIX1
02A4    26  DC              BNE    MT.ERF
02A6    EE  03              LDX    FIX1A,X
02A8    DF  00G             STX    RECCNT,D      ; STORE RESULT IN RECCNT
02AA    DE  00G             LDX    A.STRT,D      ; GET POINTER TO BEGINNING OF BUFFER
02AC    E6  08              LDA B  8.,X          ; SET UP COMPLETE MTSTATUS
02AE    C1  41              CMP B  'A,I          ; FIND DATA TYPE
02B0    27  10              BEQ    MTASCI
02B2    C1  42              CMP B  'B,I          ; IS IT BINARY
02B4    27  08              BEQ    MTBINY
02B6    C1  4E              CMP B  'N,I          ; IS IT NEW?
02B8    26  0C              BNE    MTPRG.        ; THEN MUST BE LAST FILE
02BA    86  10              LDA A  MTSNEW,I
02BC    20  06              BRA    MTPRGT
02BE    86  04      MTBINY: LDA A  MTSBIN,I      ; BINARY TYPE
02C0    20  02              BRA    MTPRGT
02C2    86  08      MTASCI: LDA A  MTSASC,I      ; ASCII TYPE
02C4    9A  00G     MTPRGT: ORA A  MTSTAT,D      ; ADD INFO TO STATUS
```

```
                02C6  E6  10      MTPRG:  LDA B   16.,X           ; GET PROGRAM TYPE
                02C8  C1  50              CMP B   'P,I            ; PROGRAM TYPE?
                02CA  27  02              BEQ     MTPRG
                02CC  20  02              BRA     MTSECT
                02CE  8A  02      MTPRG:  ORA A   MTSPGM,I        ; PROGRAM FILE
                02D0  E6  1A      MTSECT: LDA B   26.,X           ; IS IT SECRET?
                02D2  C1  53              CMP B   'S,I
                02D4  26  02              BNE     MTFDON
                02D6  8A  01              ORA A   MTSSEC,I        ; SECRET
                02D8  D6  00G     MTFDON: LDA B   MTSREG,D        ; SEE IF SHORT REC.
                02DA  C5  01              BIT B   MTFLEN,I
                02DC  27  02              BEQ     1$
                02DE  8A  80              ORA A   MTFSIZ,I
                02E0  8A  20      1$:     ORA A   MTOPEN,I                ; MARK FILE OPEN AND SAVE STATUS
                02E2  97  00G             STA A   MTSTAT,D
                02E4  BD  0435'           JSR     MTBINT          ; SET UP BUFFER POINTERS
                02E7  39                  RTS
```

```
                                        .SBTTL  MTREAD--MAGTAPE READ ROUTINE
                                ;THIS ROUTINE READS A MAGTAPE RECORD 10 TIMES BEFORE CAUSING A FATAL
                                ;ERROR.
                                        .GLOBL  MTREAD
                                        .GLOBL  DELY1S          ; 40 US DELAY ROUTINE
                                        .GLOBL  MTRDER          ; LOCATION TO KILL IF DON'T WANT FATAL ERROR
                                        .GLOBL  ZX
                                        .GLOBL  MTERR           ; MAGTAPE ERROR COUNTER
02E8  86  0A    MTREAD: LDA A   10.,1    ; GET COUNTER
02EA  36                PSH A
02EB  7F  0000G         CLR     MTERR
02EE  20  07            BRA     MTR.RR          ; SKIP BACK UP OF RECORD
02F0  36        MTR.RD: PSH A           ; SAVE IT
02F1  BD  0000G         JSR     BAKABS          ; BACKUP FOR RE-READ
02F4  7C  0000G         INC     MTERR
02F7  DE  00G   MTR.RR: LDX     A.STRT,D        ; INITIALIZE POINTERS
02F9  DF  00G           STX     MTPTR,D
02FB  DE  00G           LDX     A.MAX,D
02FD  08                INX
02FE  08                INX
02FF  DF  00G           STX     MTMAX,D
0301  BD  0300G         JSR     READRS          ; READ RECOORD
0304  BD  0697'         JSR     MTWAIT
0307  BD  0000G         JSR     PUPTAP
030A  86  0A            LDA A   10.,1           ; WAIT 400US
030C  BD  0000G         JSR     DELY1S
030F  D6  00G           LDA B   MTSTT2,D        ; CLEAR GAP BIT
0311  C4  7F            AND B   -1-MTFFST,I
0313  D7  00G           STA B   MTSTT2,D
0315  B6  0001G         LDA A   PIALY+1         ; LOOK TO SEE IF AT EOF
0318  85  40            BIT A   64.,I
031A  27  08            BEQ     1$
031C  DE  00G           LDX     ZX,D            ; ZERO VALID REG. COUNT
031E  DF  00G           STX     RECCNT,D
                        ; SET IN GAP BIT
0320  CA  80            ORA B   MTFFST,I
0322  D7  00G           STA B   MTSTT2,D
0324  7C  0001G         INC     FILLOC+1
0327  96  00G   1$:     LDA A   MTSREG,D
0329  85  02            BIT A   MTCKSM,I        ; IF NO CHECK SUM THEN GO
032B  27  09            BEQ     MTCKIT          ; GO DO CHECK SUM
032D  DE  00G           LDX     MTPTR,D         ; TEST HOW MUCH READ
032F  09                DEX                     ; REMEMBER A.MAX IS 1 GREATER FOR INPUT FUNCS.
0330  9C  00G           CPX     A.MAX,D         ; MUST COMPARE EXACTLY
0332  27  16            BEQ     MTRDON
0334  20  07            BRA     MTRCON
0336  BD  0461'  MTCKIT: JSR     MTCHKS
0339  E1  01            CMP B   1,X             ; CHECK THE CHECK SUM
033B  27  0D            BEQ     MTRDON
033D  96  00G   MTRCON: LDA A   ERRCD,D
033F  26  09            BNE     MTRDON
0341  32                PUL A
0342  4A                DEC A                   ; STILL ALLOWED TO TRY MORE READS?
0343  26  AB            BNE     MTR.RD
0345  86  00G   MTRDER: LDA A   ERMTRD,I        ;SET READ ERROR
0347  97  00G           STA A   ERRCD,D
0349  39                RTS
```

MTREAD--MAGTAPE READ ROUTINE

```
    58          034A    32                  MTRDON: PUL A
    59          034B    39                          RTS
```

```
                                        .SBTTL  MTNULL--NULL INTERNAL MAGTAPE BUFFER
                                ;THIS ROUTINE NULL FILLS FROM X TO R0 (NOT INCLUSIVE)
                                ;ENTER WITH FILL CHARACTER IN ACC. A
                                ;
                                        .GLOBL  MTNULL
034C    DE      000     MTNULL: LDX     A.STRT,D        ; START AT A.STRT
034E    A7      00      1$:     STA A   0,X
0350    08                      INX
0351    9C      000             CPX     A.MAX,D         ; DONE?
0353    26      F9              BNE     1$
0355    A7      00              STA A   0,X
0357    39                      RTS
```

```
                                        .SBTTL  MTCLOS--CLOSES MAGTAPE
                                        .GLOBL  MTCLOS
                                ;THIS ROUTINE WRITES THE CURRENT BUFFER ONTO THE MAGTAPE IF
                                ;THE CURRENT BUFFER HAS HAD ANY NEW DATA.
                                ;
0358  96  000   MTCLOS: LDA A   MTSTT2,D              ; GET STATUS TWO
035A  85  02            BIT A   MTSWRT,I        ; HAS BEEN WRITTEN ON
035C  27  17            BEQ     MTCDON
                                ;PREFORM CLOSE FUNCTION HERE <<<<<<<=========================================
035E  96  000           LDA A   PNDEOF,D                ; SAVE EOF CONDITION
0360  36                PSH A
0361  DE  000           LDX     MTPTR,D         ; POINTER TO NEXT FREE SPACE
0363  86  FF            LDA A   255.,I          ; GET EOF MARK
0365  A7  00            STA A   0,X             ; STORE IT
0367  BD  043E'         JSR     MTBSWP          ; SWAP IN PROPER BUFFERS
036A  BD  037F'         JSR     MTWRIT          ; OUTPUT THIS LAST RECORD
036D  30                TSX                     ; CLEAN UP STACK
036E  BD  0000G         JSR A9X
0371  35                TXS
0372  32                PUL A
0373  97  000           STA A   PNDEOF,D        ; RESTORE EOF CONDITIONS
0375  96  000   MTCDON: LDA A   MTSTT2,D        ; TURN OF ALL BUT FIRST REC. OUNDER HEADS
0377  84  80            AND A   #IFEST,I
0379  97  000           STA A   MTSTT2,D
037B  7F  0000G         CLR     MTSTAT
037E  39                RTS
```

```
                                        .SBTTL  MTWRIT--MAGTAPE OUTPUT ROUTINE FOR I/O
                                ;THIS ROUTINE PREFORMS THE OUTPUT OF A MAGTAPE BUFFER TO THE
                                ;3M MAGTAPE WHEN REQUESTED BY THE I/O CONTROLLER.
                                        .GLOBL  MTPOUT
                                        .GLOBL  WREND           ; DOES SOME CLEAN UP AFTER FINISHING WRRS
                                        .GLOBL  EROVRD
                                        .GLOBL  MTWRIT
                                        .GLOBL  MTFLGS          ; HARDWARE FLAGS
                    0002                MTBGFL  =  H02          ; FIRST RECORD WRITE FLAG
                                ;
 037F   DE   00G        MTWRIT: LDX     RECCNT,D                ; CAN WE REALLY OUTPUT THIS RECORD?
 0381   09                 1$:  DEX                             ; DEC. ALLOWABLE COUNT
 0382   DF   00G                STX     RECCNT,D
 0384   26   00                BNE     MTWRT                   ; IF ON LAST ON DO AN IMLICIT CLOSE
 0386   DE   00G                LDX     A.MAX,D
 0388   86   FF                 LDA A   255.,I
 038A   A7   00                 STA A   0,X
 038C   7F   0000G              CLR     MTSTT2
 038F   86   01                 LDA A   1,I                     ; SET EOF MARK
 0391   97   00G                STA A   MTFLGS,D
 0393   86   0000G      MTWRT:  LDA A   PIAMTA                  ; IS IT WRITE PROTECTED
 0396   85   10                 BIT A   20,I
 0398   27   05                 BEQ     2$
 039A   86   00G                LDA A   ERWRT,I
 039C   97   00G                STA A   ERRCD,D
 039E   39                       RTS
 039F   BD   0461'      2$:     JSR     MTCHKS                  ; GENERATE CHECKSUM
 03A2   96   00G                LDA A   MTSREG,D                ; NEED CHECKSUM?
 03A4   85   02                 BIT A   MTCKSM,I
 03A6   26   03                 BNE     3$
 03A8   08                       INX                             ; SET CHECKSUM
 03A9   E7   00                 STA B   0,X
 03AB   DF   00G        3$:     STX     MTMAX,D
 03AD   96   00G                LDA A   MTSTT2,D                ; IF FIRST RECORD UNDER HEADS DO SOMETHING
 03AF   85   80                 BIT A   MTFEST,I
 03B1   27   07                 BEQ     1$
 03B3   4F                      CLR A
 03B4   97   00G                STA A   MTSTT2,D                ; CLEAR IT
 03B6   86   02                 LDA A   MTBGFL,I
 03B8   97   00G                STA A   MTFLGS,D                ; SET HARDWARE FLAG
 03BA   DE   00G        1$:     LDX     A.STRT,D                ; OUTPUT IT TO MAGTAPE
 03BC   DF   00G                STX     MTPTR,D
 03BE   BD   0000G              JSR     WRRS                    ; WRITE IT
 03C1   BD   0697'              JSR     MTWAIT
 03C4   BD   0000G              JSR     WREND
 03C7   BD   0000G              JSR     PUPTAP
 03CA   8D   69                 BSR     MTBINT
 03CC   DE   00G                LDX     RECCNT,D                ; GET RECORD COUNT
 03CE   26   07                 BNE     4$
 03D0   86   80                 LDA A   MTFEST,I                ; SET IN THE FILE GAP FLAG
 03D2   97   00G                STA A   MTSTT2,D
 03D4   7C   00G1G              INC     FILLOC+1
 03D7   39              4$:     RTS
```

```
                                        .SBTTL  MTPOUT--MORE OF THE SAME
                                ;THIS IS THE REAL MAGTAPE OUTPUT CONTROLLER FOR BASIC
                                ;ENTER WITH:
                                ;       MTPTR   POINTING TO NEXT VALID CHAR.
                                ;       MTMAX   POINTING TO LAST VALID CHAR. POSITION
                                ;       A.PTR   POINTING TO FIRST CHAR. TO XFER
                                ;       A.END   POINTING TO LAST CHAR. TO XFER
                                ;NOTE:
                                ;       USES MTSTAT TO DECIDE WHAT BUFFER LENGTH TO USE
                                ;       ALWAYS USES STANDARD MAGTAPE BUFFER
                                ;
                                ;
                                        .GLOBL  MTPOUT
03D8    96      000     MTPOUT: LDA A   MTSTT2,D        ; TEST IF VALID INPUT BUFFER
03DA    85      10              BIT A   MTSRD,I
03DC    27      19              BEQ     MTPOT
03DE    84      EF              AND A   #-1-MTSRD,I     ; CLEAR VALID BUFFER FALG
03E0    97      000             STA A   MTSTT2,D
03E2    BD      00000G          JSR     BAKARS          ; AND BACKUP TO REWRITE RECORD
03E5    DE      000             LDX     RECCNT,D
03E7    08                      INX
03E8    DF      000             STX     RECCNT,D
03EA    B6      00000G          LDA A   PIALY           ; SET IF SLIPPED INTO A FILE GAP
03ED    85      40              BIT A   64.,I
03EF    27      06              BEQ     MTPOT
03F1    96      000             LDA A   MTSTT2,D
03F3    8A      80              ORA A   MTFFST,I
03F5    97      000             STA A   MTSTT2,D
03F7    DE      000     MTPOT:  LDX     RECCNT,D
03F9    26      05              BNE     1$
03FB    86      80              LDA A   #80.,I
03FD    97      000             STA A   PNDEOF,D
03FF    39                      RTS
0400    96      000     1$:     LDA A   MTSTT2,D
0402    8A      02              ORA A   MTSWRT,I
0404    97      000             STA A   MTSTT2,D        ; MARK BUFFER WRITTEN
0406    DE      000             LDX     A.PTR,D
0408    A6      00              LDA A   0,X             ; GET CHAR
040A    DE      000             LDX     MTPTR,D
040C    08                      INX                     ; UPDATE MAGTAPE POINTER
040D    DF      000             STX     MTPTR,D
040F    09                      DEX
0410    A7      00              STA A   0,X             ; STORE CHAR.
0412    9C      000             CPX     MTMAX,D         ; AT END YET?
0414    26      13              BNE     2$
]]]]]
0416    8D      26              BSR     MTBSWP          ; SWAP BUFFER POINTERS
0418    BD      037F            JSR     MTWRIT          ; WRITE RECORD
041B    CE      00000G          LDX     A.STAT,I        ; RESTORE OLD BUFFERS
041E    BD      00000G          JSR     PULFPN
0421    96      000             LDA A   ERRCD,D         ; ERORS DURING LAST TRANSFER?
0423    26      0A              BNE     5$
0425    96      000             LDA A   PNDEOF,D
0427    28      06              BMI     5$
0429    DE      000     2$:     LDX     A.PTR,D         ; DONE?
042B    9C      000             CPX     A.END,D
042D    26      01              BNE     4$
042F    39              5$:     RTS
```

MTPOUT--MORE OF THE SAME

```
58      0430    08              4$:     INX
59      0431    DF      00G             STX     @ PTR,0
60      0433    20      C2              BRA     MTPOT
```

```
                                        .SBTTL  MTBINT--MTBSWP--UTILITY ROUTINES
                                        .GLOBL  MTBINT
                                        .GLOBL  FRET
                                        .GLOBL  MTBSWP
                                ;MTBINT  THIS ROUTINE INITS MAGTAPE POINTERS USING A XXXX
0435    DE      00G     MTBINT: LDX     A.STRT,D
0437    DF      00G             STX     MTPTR,D
0439    DE      00G             LDX     A.MAX,D
043B    DF      00G             STX     MTMAX,D
043D    39                      RTS
                        ;
                        ;
                        ;MTBSWP  THIS ROUTINE SWAPS OUT PRESENT BUFFER CONTROL BLOCK
                        ;       AND USES MAGTAPE CONTROL BLOCK
                        ;
043E    32              MTBSWP: PUL A                   ; STASH RETURN ADDRESS
043F    97      00G             STA A   FRET,D
0441    32                      PUL A
0442    97      01G             STA A   FRET+1,D
0444    CE      0000G           LDX     A.STAT,I        ; PUSH OLD STATUS
0447    BD      0000G           JSR     PSHFPN
044A    CE      0000G           LDX     MTBFR,I         ; SET UP NEW STATUS
044D    DF      00G             STX     A.STRT,D
044F    96      00G             LDA A   MTSTAT,D
0451    2A      05              BPL     1$
0453    CE      002FG           LDX     MTBFR+127.,I
0456    20      03              BRA     2$
0458    CE      00FFG   1$:     LDX     MTBFR+255.,I
045B    DF      00G     2$:     STX     A.MAX,D
045D    DE      00G             LDX     FRET,D
045F    6E      00              JMP     0,X
```

```
                                        .SBTTL  MTCHKS--CHECK SUM COMPUTER
                                ;THIS ROUTINE GENERATES CHECKSUM WITH END AROUND CARRY
                                ;EXIT:  WITH COMPUTED SUM IN ACC B.
                                ;       ACC X IS POINTING TO LAST DATA BYTE ( A.MAX )
                                ;
0461    C6      0F      MTCHKS: LDA B   #0F,I
0463    DE      00G             LDX     A.STRT,D        ; START AT A.STRT
0465    EB      0F      1$:     ADD B   0,X             ; SIMPLE SUM
0467    C9      0L              ADC B   #0,I
0469    9C      00G             CPX     A.MAX.D         ; DONE?
046B    27      03              BEQ     CHKDON
046D    08                      INX
046E    20      F5              BRA     1$
0470    39              CHKDON: RTS
```

```
                                        .SBTTL  MAGTAPE ADDRESSING SEQUENCE
                                ;THIS SECTION ADDRESSES THE INTERNAL MAGTAPE FOR ALL FUNCTIONS
                                ;AND PREFORMS HEADER CHECKS AND UPDATES WHERE APPROPRIATE.
                                ;IT IS CALLED FROM ADRDEV WHEN A MAGTAPE (INTERNAL) OPERATION IS ENVOKED
                                ;
                                ;FIRST FOR SOME INTERPRETERS
                                ;
                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;MTPINW:        MAGTAPE WRITE TESTS
                                ;       RETURN + 2      IF NOT OPEN + LAST FILE (SETS ERROR)
                                ;       RETURN + 4      IF NEW FILE
                                ;       RETURN + 6      IF ASCII FILE
                                ;       RETURN + 8      IF BINARY FILE
                                ;
                                ;MTPINR:        MAGTAPE READ TESTS
                                ;       RETURN + 2      IF NOT OPEN + LAST FILE + SECRET + WRITTEN FILE
                                ;                               (SETS ERROR)
                                ;       RETURN + 4      IF ASCII
                                ;       RETURN + 6
                                ;
                                        .GLOBL  MTPINW
                                        .GLOBL  MTPINR
                                ;
0471    5F                      MTPINW: CLR B
0472    96      006                     LDA A   MTSTAT,D        ; GET MAGTAPE STATUS
0474    85      20                      BIT A   MTOPEN,I        ; OPEN?
0476    27      7C                      BEQ     MTRERX
0478    85      1C                      BIT A   MTSLST,I        ; LAST?
047A    27      78                      BEQ     MTRERX
047C    5C                              INC B
047D    85      10                      BIT A   MTSNEW,I        ; NEW?
047F    26      06                      BNE     MTEXX
0481    5C                              INC B
0482    85      08                      BIT A   MTSASC,I        ; ASCII?
0484    26      01                      BNE     MTEXX
0486    5C                              INC B
0487    58              MTEXX:  ASL B                   ; DO OFFSET TO RETURN ADDRESS ON STACK
0488    4F                      CLR A
0489    30                      TSX
048A    EB      01              ADD B   1,X
048C    E7      01              STA B   1,X
048E    A9      00              ADC A   0,X
0490    A7      00              STA A   0,X
0492    39                      RTS
```

```
                                        .SBTTL  MTPINR--INPUT INTERPRETER
                                ;       SEE EXPLANATION PAGE PREVIOUS
                                ;
0493  8D  0471'       MTPINR:   JSR     MTPINW        ; CALL WRITE INTERPRETER
0496  20  26                    BRA     1$            ; ERROR EXIT
0498  20  5A                    BRA     MTRERX        ; SET ERROR EXIT
049A  20  0A                    BRA     2$            ; ASCII
049C  96  000                   LDA A   MTSTT2,D      ; BINARY
049E  85  02                    BIT A   MTSWRT,I
04A0  26  52                    BNE     MTRERX        ; ERROR-ALREADY WRITTEN
04A2  C6  04                    LDA B   4,I
04A4  20  0E                    BRA     3$            ; EXIT OK
04A6  96  000           2$:     LDA A   MTSTT2,D      ; ASCII
04A8  85  02                    BIT A   MTSWRT,I
04AA  26  48                    BNE     MTRERX
04AC  96  000                   LDA A   MTSTAT,D      ; SEE IF SECRET
04AE  85  01                    BIT A   MTSSEC,I
04B0  26  42                    BNE     MTRERX
04B2  C6  02                    LDA B   2,I
04B4  4F                3$:     CLR A
04B5  30                        TSX
04B6  E8  01                    ADD B   1,X           ; OFFSET CALC
04B8  E7  01                    STA B   1,X
04BA  A9  00                    ADC A   0,X
04BC  A7  00                    STA A   0,X
04BE  39                1$:     RTS
```

```
                                        .SBTTL  THE VALIDITY DRIVERS
                                ;***************************************************
                                ;       WRITE
                                ;
0001                    0046F   80  0471'   MTPWR1: JSR     MTPINW1         ; INTERP CALL
0002                    04C2    20  72              BRA     MTEXXR          ; ERROR CODE SET
0003                    04C4    20  08              BRA     1$              ; NEW HEADER
0004                    04C6    20  2C              BRA     MTRERX          ; ERROR
0005                    04C8    96  00G             LDA A   MTSTAT,D
0006                    04CA    85  40              BIT A   MTFPOS,I
0007                    04CC    26  68              BNE     MTEXXK
0008                    04CE    96  00G     1$:     LDA A   MTSTAT,D        ; MARK STATUS APPROPRIATLY
0009                    04D0    84  EF              AND A   -1-MTSNEW,I
0010                    04D2    8A  04              ORA A   MTSBIN,I
0011                    04D4    97  00G             STA A   MTSTAT,D
0012                    04D6    CE  0066'           LDX     TYPBIN,I        ; MARK IT BINNARY/DATA
0013                    04D9    20  42              BRA     MIDATM
                                ;***************************************************
                                ;       OLD
                                ;
0014                    04D8    80  0471'   MTPOLD: JSR     MTPINW          ; WRITE INTERPRETER
0015                    04DE    20  0C              BRA     1$              ; ERROR EXIT
0016                    04E0    20  12              BRA     MTRERX          ; ERROR EXIT
0017                    04E2    20  02              BRA     2$
0018                    04E4    20  0E              BRA     MTRERX          ; ERROR EXIT
0019                    04E6    96  00G     2$:     LDA A   MTSTI2,D        ; IF WRITTEN THEN SET ERROR
0020                    04E8    85  02              BIT A   MTSWRT,I
0021                    04EA    26  08              BNE     MTRERX
0022                    04EC    39          1$:     RTS
                                ;***************************************************
                                ;       INPUT
                                ;
0023                    04ED    80  0493'   MTPINP: JSR     MTPINR          ; INPUT INTERPRETER
0024                    04F0    20  06              BRA     MTP1            ; ERROR EXIT
0025                    04F2    20  04              BRA     MTP1
0026                    04F4    86  00G     MTRERX: LDA A   ERMILA,I        ; MAGTAPE ACCESS ERROR
0027                    04F6    97  00G             STA A   ERRCD,D
0028                    04F8    39          MTP1:   RTS
                                ;***************************************************
                                ;       READ
                                ;
0029                    04F9    80  0493'   MTPRD:  JSR     MTPINR          ; INPUT INTERP.
0030                    04FC    20  02              BRA     1$              ; ERROR EXIT
0031                    04FE    20  F4              BRA     MTRERX          ; ERROR
0032                    0500    39          1$:     RTS                     ; OK
                                ;***************************************************
                                ;       PRINT
                                ;
0033                    0501    80  0471'   MTPPR1: JSR     MTPINW  ; INTERP CALL
0034                    0504    20  30              BRA     MTEXXR          ; ERROR EXIT
0035                    0506    20  0A              BRA     2$              ; MARK NEW HEADER
0036                    0508    20  02              BRA     1$
0037                    050A    20  E8              BRA     MTRERX          ; ERROR EXIT
0038                    050C    96  00G     1$:     LDA A   MTSTAT,D
0039                    050E    85  40              BIT A   MTFPOS,I
0040                    0510    26  24              BNE     MTEXXK
0041                    0512    96  00G     2$:     LDA A   MTSTAT,D
```

```
                                AND A   -1-MTSNEW,I
                                ORA A   MTSASC,I
                                STA A   MTSTAT,D
                                LDX     TYPASC,I        ; MARK IT ASCII/DATA
                MTDATM:         JSR     MOOT88
                                LDX     PRGDAT,I
                                LDA A   10.,I
                                LDA B   16.,I
                MTEXXQ:         JSR     MOOMTH
                MTEXXZ:         LDA A   MTSREG,D        ; SEE IF OK TO ASSUME A HEADER
                                BIT A   MTNHDR,I
                                BNE     MTEXXR
                                JSR     MFR8            ; THEN BACK UP TO WHRITE IT
                                JSR     MTWRT
                MTEXXK:
                MTEXXR:         RTS
                ;***************************************************************
                ;       SAVE
                ;
                MTPSAV:         JSR     MTPINW          ; INTERP. CALL
                                BRA     MTEXXR          ; ERROR EXIT
                                BRA     2$              ; MARK HEADER
                                BRA     1$
                                BRA     MTRERX          ; ERROR EXIT
                1$:             LDA A   MTSTAT,D
                                BIT A   MTEPOS,I
                                BNE     MTEXXK
                                BIT A   MTSSEC,I
                                BNE     MTRERX
                2$:             LDA A   XAXIS,D
                                BNE     MTRERX
                                LDX     TYPASC,I        ; ASCII/ PROGRAM
                                JSR     MOOT88
                                LDX     PRGPGM,I
                                LDA A   10.,I
                                LDA B   16.,I
                                JSR     MOOMTH
                                LDA A   MTSTAT,D        ; MARK PROPER HEADER
                                AND B   -1-MTSNEW,I
                                ORA B   MTSASC+MTSPGM,I
                                STA B   MTSTAT,D
                                LDA A   STAT37,D
                                BEQ     MTEXXZ
                                LDX     TYPSEC,I
                                LDA A   8.,I
                                LDA B   26.,I
                                BRA     MTEXXQ
```

```
                                        .SBTTL  MTPIN---MAGTAPE INPUT HAND SHAKE UNDER BASIC
                                        ;THIS STUFF DOES WHAT EVER IT IS SUPPOST TO DO?
                                        ;ENTER WITH:
                                        ;       MTPTR   POINTING TO FIRST VALID CHARACTER IN MAGTAPE BUFFER
                                        ;       MTMAX   POINTING TO LAST VALID CHARACTER POSITION IN BUFFER
                                        ;                       (EVEN THOUGH MAY HAVE EOF BEFORE THEN)
                                        ;       A.PTR   POINTING TO FIRST POSITION IN RECIEVING BUFFER
                                        ;       A.MAX   POINTING TO LAST POSITION IN RECIEVING BUFFER
                                        ;EXIT WITH:
                                        ;       A.END   POINTING TO LAST XMITTED CHARACTER + 1
                                        ;               (EXITS WHEN ENCOUNTER "CR" OR A.MAX)
                                        ;STATUS BYTES:
                                        ;       MTSTT2  MTSRD IS SET IF BUFFER IS VALID FOR INPUT
                                        ;       CRSTAT  CREOT IS SET IF EOF IS ENCOUNTERED ON MAGTAPE
                                        ;               (A "CR" WILL BE PLACED IN BUFFER)
                                        ;
                                        ;
                                                .GLOBL  MTPIN,MTRBFR
                                                .GLOBL  SCRMS
                                                .GLOBL  ETXCHR,EOLCHR
                                                .GLOBL  NULCHR
                                        ;
0575  96  000   MTPIN:  LDA A   MTSTT2,D     ; VALID BUFFER?
0577  85  10            BIT A   MTSRD,I
0579  26  2A             BNE    RDBFRV       ; YES
057B  BD  0583'          JSR    MTRBFR       ; GET NEW BUFFER
057E  96  000            LDA A  ERRCD,D
0580  27  23             BEQ    RDBFRV
0582  39                RTS
                        ;xxxxxxxxxxREAD NEW MAGTAPE BUFFER INTO MTBFR xxxxxxxxxx
                        ;
0583  DE  000   MTRBFR: LDX     RECCNT,D     ; LOOK AT VALID REC. COUNT
0585  27  64             BEQ    RDEOF
0587  09                DEX
0588  DF  000           STX     RECCNT,D
058A  BD  043E'          JSR    MTBSWP       ; READ IT
058D  BD  02E8'          JSR    MTREAD
0590  BD  0435'          JSR    MTBINT       ; RESET BUFFER POINTERS
0593  CE  0000G          LDX    A.STAT,I
0596  BD  0000G          JSR    PULFPN
0599  96  000            LDA A  ERRCD,D      ; JANY READ ERRORS?
059B  27  01             BEQ    1$
059D  39                 RTS
059E  96  000   1$:     LDA A   MTSTT2,D
05A0  8A  10            ORA A   MTSRD,I
05A2  97  000           STA A   MISTT2,D
05A4  39                RTS
                        ;
                        ;
05A5  DE  000   RDBFRV: LDX     MTPTR,D      ; GET CHAR FROM BUFFER
05A7  A6  00            LDA A   0,X
05A9  D6  000           LDA B   IOFUNC,D     ; DIFERENT IF DOING READ
05AB  C1  0E            CMP B   14.,I
05AD  27  07            BEQ     5$
05AF  91  000           CMP A   ETXCHR,D     ; EOF?
05B1  27  38            BEQ     RDEOF
05B3  BD  0000G         JSR     SCRMS
```

MTPIN---MAGTAPE INPUT HAND SHAKE UNDER BASIC

```
    58  0586  08            5$:   INX                 ; UPDATE MTPTR
    59  0587  DF  00G             STX   MTPTR,D
    60  0589  09                  DEX
    61  058A  9C  00G             CPX   MTMAX,D       ; LAST CHAR?
    62  058C  26  06              BNE   1$
    63  058E  D6  00G             LDA B MTSTT2,D
    64  05C0  C4  EF              AND B -1-MTSRD,I
    65  05C2  D7  00G             STA B MTSTT2,D
    66
    67  05C4  D6  00G       1$:   LDA B NULCHR,D      ; IGNORE NUL CHAR.S
    68  05C6  28  03              BMI   4$
    69  05C8  11                  CBA
    70  05C9  27  AA              BEQ   MTPIN
    71  05CB  DE  00G       4$:   LDX   A.PTR,D       ; SAVE OUTPUT CHAR
    72  05CD  A7  00              STA A 0,X
    73  05CF  D6  00G             LDA B IOFUNC,D      ; DOING A READ?
    74  05D1  C1  0E              CMP B 14.,I
    75  05D3  27  04              BEQ   2$
    76  05D5  91  00G             CMP A EOLCHR,D      ; FOUND EOL?
    77  05D7  27  09              BEQ   RDBCRE
    78  05D9  9C  00G       2$:   CPX   A.MAX,D       ; DONE WITH BUFFER?
    79  05DB  27  0A              BEQ   RDBEND
    80  05DD  08                  INX                 ; UPDATE OUTPUT POINTER
    81  05DE  DF  00G             STX   A.PTR,D
    82  05E0  20  93              BRA   MTPIN         ; GO FOR NEXT CHAR
    83  05E2  86  01        RDBCRE: LDA A CRNORM,I
    84  05E4  97  00G             STA A CRSTAT,D
    85  05E6  09            RDBEN.: DEX
    86  05E7  08            RDBEND: INX
    87  05E8  DF  00G             STX   A.END,D
    88  05EA  39                  RTS                 ; EXIT
    89  05EB  86  20        RDEOF:  LDA A CREOT,I
    90  05ED  97  00G             STA A CRSTAT,D
    91  05EF  DE  00G             LDX   A.PTR,D
    92  05F1  86  0D              LDA A 15.,I         ; SET "CR" INTO BUFFER
    93  05F3  A7  00              STA A 0,X
    94  05F5  86  80              LDA A 128.,I        ; SET EOF DEVICE 0
    95  05F7  97  00G             STA A PNDEOF,D
    96  05F9  20  E8              BRA   RDBEN.        ; EXIT
```

```
                                        .SBTTL  MTSSET--MAGTAPE INTERNAL STATUS SET UP
                                ;THIS ROUTINE SETS UP THE MAGTAPE STATUS FOR TRANSFERS
                                ;MTSREG CONTAINS STATUS:
                                ;       BIT 0--256/128
                                ;       BIT 1--CHECKSUM/NO CHECKSUM
                                ;       BIT 2--HEADER/NO HEADER
                                ;
                                        .GLOBL  MTSSET
                                        .GLOBL  YAXIS
                                ;
05FB    80      0000G   MTSSET: JSR     F.IXNUM         ; GET NUMBER
05FE    96      00G             LDA A   YAXIS,D         ; RETRIEVE UNIVERSAL COUNT REGISTER
0600    26      04              BNE     1$              ; IF ALLREADY IN USE THEN NO INITIALIZE
0602    7F      0000G           CLR     MTSREG
0605    4C                      INC A
0606    48              1$:     ASL A                   ; MULTIPLY BY 2 FOR NEXT TIME
0607    97      00G             STA A   YAXIS,D
0609    44                      LSR A
060A    50                      TST B                   ; IF BYTE=0 THEN CLEAR APPROPRIATE STATUS BYTE
060B    26      05              BNE     SETBIT
060D    43                      COM A                   ; MAKE A MASK
060E    94      00G             AND A   MTSREG,D
0610    20      02              BRA     MTSEX1
0612    9A      00G     SETBIT: ORA A   MTSREG,D        ; SET BIT
0614    97      00G     MTSEX1: STA A   MTSREG,D
0616    D6      00G     MTSEX:  LDA B   YAXIS,D         ; SEE IF NEED TO END THIS NONSENSE
0618    C1      04              CMP B   4,I
061A    23      04              BLS     1$
061C    86      12              LDA A   18.,I           ; THIS TERMINATES IOPROC
061E    97      00G             STA A   IOFLGS,D
0620    39              1$:     RTS
```

```
                                        .SBTTL  MTSCRT--MAGTAPE SECRET CALL
                                        .GLOBL  MTSCRT
                                ;THIS ROUTINE SETS THE MAGTAPE TO DO SECRET ON THE NEXT SAVED FILE
                                ;
0621    96      00G     MTSCRT: LDA A   MTSTAT,D
0623    8A      01              ORA A   MTSSEC,I
0625    97      00G             STA A   MTSTAT,D
0627    39              15:     RTS
```

```
                                        .SBTTL  MTPADR--ACTUAL ADDRESSING CALL
                                ;THIS ROUTINE IS CALLED FROM ADRDEV TO ADDRESS THE MAGTAPE
                                ;FOR THE OPERATIONS OF PRINT, OLD, SAVE, INPUT, READ, WRTIE
                                ;
                                        .GLOBL  MTPADR
                                        .GLOBL  PIAHX,PIALX,PIAHY,PIALY
                                        .GLOBL  PIAMTA,PIAMTB
                                        .GLOBL  DSPSIT
                                        .GLOBL  PUPTAF
                                ;
0628    8D      22              MTPADR: BSR     INITMT          ; INIT MT PIA'S
062A    96      00G                     LDA A   ERRCD,D
062C    26      0A                      BNE     1$
062E    8D      0000G                   JSR     BFRALC          ; ALLOCATE BUFFER FOR PLAYING
0631    96      00G                     LDA A   A.SEC,D         ; GET ADDRESS FUNCTION
0633    CE      0639'                   LDX     MTPSPC,I        ; SPECIAL FUNCTION TABLE
0636    8D      78                      BSR     XFRSCN          ; SPECIAL SCANNER
0638    39              1$:             RTS
                                ;
                                ;       SPECIAL TABLE
                                ;
                                        .MACRO  SCN K,KADR
                                        .BYTE K
                                        .WORD   KADR
                                        .ENDM
                                ;
0639                            MTPSPC: SCN     12.,MTPPRI      ; PRINT
063C                                    SCN     4.,MTPOLD       ; OLD
063F                                    SCN     14.,MTPRD       ; READ
0642                                    SCN     13.,MTPINP      ; INPUT
0645                                    SCN     15.,MTPWRI      ; WRITE
0648                                    SCN     1.,MTPSAV       ; SAVE
0648    00                              .BYTE 0                 ; END OF TABLE
```

```
                                      .SBTTL  INITMT--INITIALIZE MAG TAPE PIA'S
                                      .GLOBL  PIAMTA
                                      .GLOBL  PIAHY
                                      .GLOBL  ERNCRT
                                      .GLOBL  ERWRT
                                      .GLOBL  INITMT
                                      .GLOBL  DRBUSY
    064C   BD   0000G       INITMT:   JSR     DRBUSY          ; WAIT FOR DISPLAY
    064F   D6     00G                 LDA B   DSPSTT,D
    0651   2B     3B                  BMI     M.11
    0653   BD   0000G                 JSR     DISBLE
    0656   C6     80                  LDA B   128.,I
    0658   D7     00G                 STA B   DSPSTT,D
    065A   C6     86                  LDA B   206,I           ; CENTER SCREEN
    065C   F7   0000G                 STA B   PIALY
    065F   C6     01                  LDA B   1,I
    0661   F7   0000G                 STA B   PIAHY
    0664   C6     FC                  LDA B   376,I
    0666   F7   0000G                 STA B   PIAHX
    0669   7F   0000G                 CLR     PIALX
    066C   BD   0000G                 JSR     PUPTAP          ; INITIALIZE MAGTAPE
                                      .GLOBL  DISBLE
                                ;TEST MAGTAPE SWITCHES HERE<<<<<<<<<<<<<<<<<<<<========  ============================
    066F   B6   0000G                 LDA A   PIALT           ;CHANGED BY ROGER THE LODGER
    0672   85     08                  BIT A   10,I            ;DITTO
    0674   27     18                  BEQ     M.11
    0676   B6   0000G                 LDA A   PIAMTA
    0679   88     02                  EOR A   2,I             ; CLEAR FLIP-FLOP
    067B   B7   0000G                 STA A   PIAMTA
    067E   88     02                  EOR A   2,I
    0680   B7   0000G                 STA A   PIAMTA
    0683   B6   0000G                 LDA A   PIALT           ; IS CARTRIGE THERE NOW?
    0686   85     08                  BIT A   10,I            ;THIS AND PREVIOUS INSTR. ALSO
                                                              ;WERE CHANGED BY ROGER THE LODGER
    0688   27     05                  BEQ     M.22
    068A   86     00G                 LDA A   ERNCRT,I
    068C   97     00G                 STA A   ERRCD,D
    068E   39               M.11:     RTS
    068F   5F               M.22:     CLR B                   ; NEW CART. SET UP
    0690   D7     00G                 STA B   MTSTT2,D
    0692   D7     00G                 STA B   MTSTAT,D
    0694   7E   0000G                 JMP     REWIND
```

```
                                        .SBTTL  MTWAIT--WAIT FOR MAGTAPE ROUTINE
                                        .GLOBL  PIALT
0697  96  00G       MTWAIT: LDA A   MAGEND,0        ;MAGTAPE OPERATION DONE?
0699  27  14                BEQ     3$
069B  B6  0000G             LDA A   PIALT                   ;WAS CARTRIDGE REMOVED?
069E  85  08                BIT A   10,I
03A0  27  04                BEQ     1$
06A2  86  00G               LDA A   ERNCRT,I
06A4  20  07                BRA     2$
06A6  B6  0001G     1$:     LDA A   PIAMTB+1                ;EOT?
06A9  2A  EC                BPL     MTWAIT
06AB  86  00G               LDA A   EREOM,I
06AD  97  00G       2$:     STA A   ERRCD,0
06AF  39            3$:     RTS
```

```
                                        .SBTTL  XFRSCN--SPECIAL TABLE SCANNER
                                ;ROUTINE PASSES CONTROL TO INDEX USING ACC A
                                ;
                                        .GLOBL  XFRSCN
                                ;TABLE FORMAT
                                ;       BYTE 1 IS COMPARISION TO ACC A
                                ;       WORD NEXT IS WHERE TO GO
                                ;
0680  60  00    XFRSCN: TST     0,X             ; AT END OF TABLE
0682  27  17            BEQ     XFRRTN
0684  A1  00            CMP A   0,X
0686  27  05            BEQ     XFRIT
0688  08                INX
0689  08                INX
068A  08                INX                     ; MOVE TO NEXT ENTRY
068B  20  F3            BRA     XFRSCN
068D  EE  01    XFRIT:  LDX     1,X
068F  AD  00            JSR     0,X             ; JUMP TO ROUTINE
06C1  86  23            LDA A   MTPD2,I         ; IF ONE OF THOSE FOUND USE SPECIAL
06C3  97  00G           STA A   A.PRIM,D        ; MAGTAPE DEVICE ADDRESS
06C5  96  00G           LDA A   MTSTAT,D        ; CLEAR FIRST ACCESS BIT
06C7  8A  40            ORA A   MTFPOS,I
06C9  97  00G           STA A   MTSTAT,D
06CB  39        XFRRTN: RTS
                ;
                ;NOTE:   THIS ROUTINE COULD BE MODIFIED TO CLR OR SET ACC A
                ;       IF IT WAS NECESSARY TO USE IT SOMEPLCE BESIDES
                ;       IN THE MAGTAPE SECTION
```

```
                                .SBTTL  TAPFIL  LIST,SAVE AND OLD MODULE
                        ;THESE ROUTINES PREFORM THE LIST SAVE AND OLD FUNCTION
                        ;GIVEN A CALL FROM THE EVALUATOR TO THE NAME TAPFIL
                        ;
                                .GLOBL  TAPFIL
                                .GLOBL  UNCOMP
                                .GLOBL  GETLAR
                                .GLOBL  ADRDEV
                                .GLOBL  UNADR
                                .GLOBL  STKBLD
                                .GLOBL  IOCLNR
                                .GLOBL  SNDBFR
                                .GLOBL  PUTBYT
                                .GLOBL  BFR%LC
                                .GLOBL  FIX1
                        ;
                        ;       SOME CONSTANTS
                        ;
0013                    LSTFNC  = 19.
0001                    SAVFNC  = 1.
```

```
                                        .SBTTL  TAPFIL  OLD, SAVE, AND LIST FUNCTION
                                ;THIS IS THE ENTRY POINT FROM THE EVALUATOR WHICH ENVOKES
                                ;ANY OF THE ABOVE MENTIONED FUNCTIONS
                                ;
                                        .GLOBL  FILERQ
                                        .GLOBL  CRLF,ERNOLD,ERLOLD,YAXIS
                                        .GLOBL  TAPF2           ; HELLO
                                        .GLOBL  OLDONE
                                        .GLOBL  NEWBFR
                                        .GLOBL  TRANSL
                                ;
                                ;*****************************************************************
                                ;       SPECIAL TRICK FOR OLD
                                ;
06CC  BD  0000G         OLDONE: JSR     DLTALL
06CF  86  00G                   LDA A   APPCOD,I        ; FAKE BETTER
06D1  97  00G                   STA A   CTKN,D
06D3  CE  0000G                 LDX     OLD,I
06D6  DF  00G                   STX     OPRADR,D
06D8  36                        PSH A
06D9  36                        PSH A
06DA  86  00G                   LDA A   EOLTG,I
06DC  36                        PSH A
06DD  BD  0768'                 JSR     TAPFIL          ; THAT OUGHT TO BE INTERESTING
06E0  32                        PUL A
06E1  32                        PUL A
06E2  32                        PUL A

06E3  96  00G                   LDA A   ERRCD,D
06E5  26  08                    BNE     1$
06E7  96  00G                   LDA A   GLBFLG,D
06E9  2A  07                    BPL     1$
06EB  DE  00G                   LDX     PGMPTR,D
06ED  DF  00G                   STX     NLPTR,D
06EF  BD  0000G                 JSR     PGMEVL
06F2  7E  0000G         1$:     JMP     IDLE
                                ;
                                ;*****************************************************************
                                ;ONE OF THOSE LITTLE SUBROUTINES TO PREFORM SOME LITTLE FUNCTION
                                ;IN MORE THAN ONE PLACE
                                ;THIS ONE GETS A INTEGER OUT OF THE MIDDLE OF THE STACK SOME HOW
                                ;
                                ; SETS ACC. B TO 1 IF NO ARG. ON STACK
                                ; AND SETS IT TO ZERO IF ARGUMENT FOUND.
06F5  32                FILINN: PUL A                   ; EXTRACT RETURN ADDRESS
06F6  97  01G                   STA A   DREXTB+1,D
06F8  32                        PUL A
06F9  97  02G                   STA A   DREXTB+2,D
06FB  32                        PUL A                   ; LOOK AT TAG
06FC  81  00G                   CMP A   EOLTG,I
06FE  26  09                    BNE     FILNNN
0700  36                        PSH A                   ; IF EOL THEN NO NUMBER
0701  CE  0000          FILINM: LDX     0,I
0704  C6  FF                    LDA B   255.,I
0706  7E  0000G                 JMP     DREXTB
0709  32                FILNNN: PUL A
070A  30                        TSX
```

```
                                                LDX     0,X             ; GET POINTER TO NUMBER IN STACK
                                                INS
                                                INS
                                                TST A
                                                BNE     FILNM           ; IF UNDEFINED THEN 0 LINENUMBER
                                                INX
                                                JSR     FIX1            ; FIX NUMBER IN STACK
                                                LDX     FIX1A,X
                                                CLR B
                                                JMP     DREXTB          ; EXIT
                                        ;*******************************************************
                                        ;       OLD DRIVER (OLD IS IN "OLD" NOT AS IN AGED)
                                        ;
                                                .GLOBL  IDLE,APPCOD,OLD,OPRADF,PGMPTR,NLPTR,PGMEVL
                                                .GLOBL  CTKN,OLDCOD,DLTALL
071C    96      00G     TYPOLD: LDA A   CTKN,D          ; TEST IF REALLY DOING AN OLD
071E    81      00G             CMP A   OLDCOD,I
0720    27      AA              BEQ     OLDONE          ; GO DO SPECIAL TRICK FOR OLD
0722    7F      0000G           CLR     YAXIS           ; ANY THING OLDED FLAG
0725    BD      0000G   TOLD:   JSR     NEWBER          ; GET NEW BUFFER FOR LEX
0728    96      00G             LDA A   ERRCD,D
072A    26      26              BNE     OLDXX
072C    96      00G             LDA A   CRSTAT,D        ; SET AN ERROR FOR LINES LONGER THAN 73 CHARS
072E    26      06              BNE     1$
0730    86      30G             LDA A   ERLOLD,I
0732    97      00G             STA A   ERRCD,D
0734    20      19              BRA     OLDXXX
0736    DE      00G     1$:     LDX     A.END,D
0738    86      00              LDA A   CR,I
073A    A7      00              STA A   0,X
073C    BD      0000G           JSR     TRANSL
073F    96      00G             LDA A   CRSTAT,D        ; AN EOF CONDITION DURING LAST BUFFER?
0741    85      3C              BIT A   EOFTYP+CRE01,I
0743    26      0A              BNE     OLDXXX
0745    96      00G             LDA A   ERRCD,D         ; ANY ERRORS FROM TRANSL
0747    26      11              BNE     OLDX
0749    86      01              LDA A   1,I             ; SET SOMETHING OLDED FLAG
074B    97      00G             STA A   YAXIS,D
074D    20      06              BRA     TOLD            ; AND GO FOR MORE
074F    7F      0000G   OLDXXX: CLR     PNOEOF
0752    96      00G     OLDXX:  LDA A   YAXIS,D         ; SET ERROR IF NOTHING OLDED
0754    26      04              BNE     OLDX
0756    86      00G             LDA A   ERNOLD,I
0758    97      00G             STA A   ERRCD,D
075A    7E      07FF'   OLDX:   JMP     LSTEXT
                                        ;*******************************************************
                                        ;       THIS HERE IS THE MAIN ENTRY POINT
                                        ;
075D    30              TAPFIL: TSX                     ; SAVE STACK POINTER FOR STKBLD
075E    08                      INX
075F    DF      00G             STX     R0,D
0761    86      00G     TAPF2:  LDA A   EOLTG,I         ; TAG IT
0763    36                      PSH A
0764    BD      0000G           JSR     STKBLD          ; GO FORM A REAL STACK
0767    30                      TSX
0768    A6      00              LDA A   0,X             ; EOL?
076A    81      00G             CMP A   EOLTG,I         ; THEN NO ARGUMENTS
```

```
  125  077A  27  0B              BEQ     TYPEXT
  126  077C  A6  01              LDA A   1,X             ; THEN GET TYPE
  127  077E  81  20              CMP A   ASTR,I          ; FILE DEVICE COMMAND?
  128  0780  26  05              BNE     TYPEXT
  129  0782  86  0C              LDA A   12.,I
  130  0784  BD  0000G           JSR     FILERQ          ; SPECIAL FILE PROCESSOR
  131  0787  96  00G     TYPEXT: LDA A   ERRCD,D
  132  0789  26  0D              BNE     OLDX
  133  078B  BD  0000G           JSR     ADPDEV          ; GO ADDRESS DEVICE AND CHECK HEADERS
  134                                                    ; IF APPROPRIATE
  135  078E  96  00G             LDA A   A.PRIM,D        ; PUT LIST,OLD,SAVE STUFF IN EDIT BUFFER
  136  0790  36                  PSH A
  137  0791  86  1F              LDA A   KBDEV,I
  138  0793  97  00G             STA A   A.PRIM,D
  139  0795  BD  0000G           JSR     BFRALC
  140  0798  32                  PUL A
  141  0799  97  00G             STA A   A.PRIM,D        ; SET THINGS BACK
  142  079B  BD  06F5'           JSR     FILINN          ; GO EXTRACT LINE NUMBERS FROM STACK
  143  079E  D7  00G             STA B   R15,D           ; SAVE STATUS BYTE
  144  07A0  DF  00G             STX     R10,D
  145  07A2  BD  06F5'           JSR     FILINN
  146  07A5  26  04              BNE     1$              ; TEST PRESENT ARG
  147  07A7  DF  00G             STX     R11,D           ; BOTH ARGS PRESENT SO PROCEED
  148  07A9  20  00              BRA     LITCNT
  149  07AB  DE  00G     1$:     LDX     R10,D           ; SECOND ARG NOT PRESENT
  150  07AD  DF  00G             STX     R11,D           ; SO USE FIRST ARG. AS LAST ARG
  151  07AF  96  00G             LDA A   R15,D           ; TEST IF BOTH ARGS. NOT PRESENT
  152  07B1  27  05              BEQ     LITCNT
  153  07B3  CE  FFFF            LDX     65535.,I        ; NEED TO LIST EVERYTHING
  154  07B6  DF  00G             STX     R11,D
  155  07B8  96  00G     LITCNT: LDA A   ERRCD,D         ; ANY ERRORS THUS FAR?
  156  07BA  26  9C              BNE     OLDX
  157
  158
  159                            IF DF REALLY
  160                            LDX     A.MAX,D
  161                            CMP A   1,I
  162                            BEQ     1$
  163                            INX
  164                    1$:     STX     A.MAX,D         ; I DON'T KNOW IF THAT WILL WORK
  165                                                    ; BUT THERE IT IS
  166                            .ENDC
  167
  168                    ;
  169                    ;       INFORMATION
]]]]]
  170                    ;R10    = STARTING LINE NUMBER
  171                    ;R11    = ENDING LINE NUMBER
  172                    ;
  173  07BC  96  00G             LDA A   IOFUNC,D
  174  07BE  81  04              CMP A   4,I             ; "OLD"?
  175  07C0  26  03              BNE     LISSOM
  176  07C2  7E  071C'           JMP     TYPOLD
  177  07C5  D6  01G     LISSOM: LDA B   R10+1,D         ; PUT LINE NUMBER ON STACK
  178  07C7  37                  PSH B
  179  07C8  D6  00G             LDA B   R10,D
  180  07CA  37                  PSH B
  181  07CB  37                  PSH B                   ; TAG IT
```

```
182  07CC  BD  0000G          JSR   GETLAR          ; GET LINE POINTER
183  07CF  33                 PUL B
184  07D0  30                 TSX
185  07D1  EE  00             LDX   0,X
186  07D3  31                 INS                   ; CLEAN UP STACK
187  07D4  31                 INS
188  07D5  DF  00G            STX   R1,D            ; SAVE FOR UNCOMP
189  07D7  26  02             BNE   1$              ; IF ZERO THEN NO MORE PROGRAM
190  07D9  20  1E             BRA   LSXNRM
191  07DB  33            1$:  PUL B                 ; ELSE STASH AWAY LINE NUMBER
192  07DC  30                 TSX
193  07DD  EE  00             LDX   0,X             ; LINE NUMBER FOUND
194  07DF  DF  00G            STX   R10,D
195  07E1  31                 INS                   ; CLEAN STACK
196  07E2  31                 INS
197  07E3  96  00G            LDA A R10,D           ; TEST IF DONE
198  07E5  D6  01G            LDA B R10+1,D
199  07E7  91  00G            CMP A R11,D
200  07E9  22  0E             BHI   LSXNRM
201  07EB  25  20             BCS   LSTAGN
202  07ED  D1  01G            CMP B R11+1,D
203  07EF  22  08             BHI   LSXNRM
204  07F1  25  27             BCS   LSTAGN
205  07F3  DE  00G            LDX   ZX,D
206  07F5  DF  00G            STX   R11,D
207  07F7  20  21             BRA   LSTAGN
208                 ;*****************************************************************
209                 ;       CLEAN UP IS HERE FOR INCONSISTANCY
210                 ;
211  07F9  BD  0000G  LSXNRM: JSR   CRLF
212  07FC  BD  0000G          JSR   SNDBFR
213  07FF             LSTEXT: LSTFIX:
214  07FF  DE  00G            LDX   ZX,D            ; SET UP CLEAN EDIT BUFFER
215  0801  DF  00G            STX   EDTMAX,D
216  0803  DF  00G            STX   EDTPTR,D
217                           .GLOBL EDTMAX
218  0805  96  00G  LSTABT:   LDA A A.PRIM,D
219  0807  81  23             CMP A MTPD2,I
220  0809  26  09             BNE   1$              ; IF MAGTAPE SAVE THEN CLOSE FILE
221  080B  96  00G            LDA A A.SEC,D
222  080D  81  01             CMP A SAVENC,I
223  080F  26  03             BNE   1$
224  0811  BD  0358'          JSR   MTCLOS
225  0814  BD  0000G  1$:     JSR   UNADR
226  0817  BD  0000G          JSR   IOCLNR          ; CLENA UP STACK
227                 ;**************
228  081A  BD  0000G  LSTAGN: JSR   UNCOMP          ; UNLEX THE LINE
229  081D  96  00G            LDA A ERRCD,D         ; DID IT FIT?
230  081F  26  DE             BNE   LSTEXT
231  0821  5F                 CLR B
232  0822  96  00G            LDA A IOFUNC,D        ; IS IT A LIST?
233  0824  81  13             CMP A LSTFNC,I        ; NEED TO SET LIST MODE TO I/O DRIVERS
234  0826  26  02             BNE   LSTSND
235  0828  CA  02             ORA B LSTFMT,I
236  082A  D7  00G  LSTSND:   STA B NOOUT,D
237  082C  BD  0000G          JSR   CRLF            ; SEND "CR"
238  082F  BD  0000G          JSR   OUTBFR
```

```
239                                     .GLOBL  OUTBFR
240     0832    7F      0000G           CLR     NOOUT
241     0835    96      00G             LDA A   PNDEOF,D
242     0837    9A      00G             ORA A   ERRCD,D
243     0839    26      C4              BNE     LSIEXT
244     083B    DE      00G             LDX     R11,D           ; IF ZERO THEN DONE
245     083D    27      BA              BEQ     LSXNRM
246     083F    DE      00G             LDX     R10,D           ; FORCE TO NEXT LINE
247     0841    08                      INX
248     0842    DF      00G             STX     R10,D
249     0844    7E      02C5'           JMP     LISSOM
```

```
                                        .SBTTL  TLIST---THE TLIST COMMAND WITH NO OPTIONALS
                                        .GLOBL  CRTRST
                                        .GLOBL  LIST,FPAITT
                                        .GLOBL  OSPOUT
                                        .GLOBL  TLIST
0847    86      000     TLIST:  LDA A   EOLTG,I         ; TAG FOR EVERYONE
0849    36                      PSH A
084A    CE      0000G           LDX     LIST,I
084D    DF      00G             STX     OPRADR,D
084F    BD      0000G           JSR     ADRDEV          ; GET OPTIONAL ADDRESS
0852    C6      01              LDA B   1,I             ; INITIALIZE LOCATION COUNTER
0854    96      00G             LDA A   MTSREG,D
0856    85      04              BIT A   MTNHDR,I
0858    26      5C              BNE     TLDON
085A    D7      00G     TLISTT: STA B   LENGTH,D
085C    96      00G             LDA A   A.PRIM,D                ; SAVE STATUS
085E    36                      PSH A
085F    86      21              LDA A   33.,I                   ; SET UP FOR MAGTAPE
0861    97      00G             STA A   A.PRIM,D
0863    BD      0000G           JSR     BFRGLC                  ; GET INPUT BUFFER
0866    BD      064C'           JSR     INITMT
0869    D6      00G             LDA B   LENGTH,D        ; FIND IT
086B    BD      021D'           JSR     MTAFIN
086E    BD      0000G           JSR     CRTRST          ; RESTORE STATUS
0871    32                      PUL A
0872    97      00G             STA A   A.PRIM,D
0874    96      00G             LDA A   ERRCD,D
0876    26      3E              BNE     TLDON
0878    CE      0021G           LDX     MTBFR+33.,I     ; CONVERT RECORDS TO BYTES!!
087B    DF      00G             STX     R0,D
087D    CE      002BG           LDX     MTBFR+43.,I
0880    DF      00G             STX     R1,D
0882    BD      0000G           JSR     AFPITT
0885    FE      0000G           LDX     FPC
0888    BD      0000G           JSR     A7X             ; INC. 7 IN EXPONENT
088B    96      00G             LDA A   MTSTAT,D        ; CHECK ON REC. LENGTH
088D    2B      01              BMI     1$
088F    08                      INX
0890    FF      0000G   1$:     STX     FPC             ; INC. 8 IN THE EXPONENT
                                .GLOBL  A7X
0893    CE      0021G           LDX     MTBFR+33.,I
0896    DF      00G             STX     R1,D
0898    CE      0014            LDX     20.,I
089B    DF      00G             STX     R2,D
089D    BD      0000G           JSR     FPAITT  ; PUT BYTE INFO IN THE BUFFER
08A0    CE      0000G           LDX     MTBFR,I
08A3    DF      00G             STX     A.PTR,D
08A5    CE      002BG           LDX     MTBFR+43.,I
08A8    DF      00G             STX     A.END,D
08AB    BD      0000G           JSR     SNDBFR          ; OUTPUT THIS MESS
08AD    D6      00G             LDA B   LENGTH,D
08AF    5C                      INC B
08B0    96      00G             LDA A   MTSTAT,D        ; WAS LAST ONE LAST FILE?
08B2    85      1C              BIT A   MTSLST,I
08B4    26      A4              BNE     TLISTT
08B6    96      00G     TLDON:  LDA A   ERRCD,D
08B8    36                      PSH A
```

```
    58          08B9      BD    0000G               JSR     MTPRWD          ; REWIND IT
    59                                                GLOBL   MTPRWD
    60          08BC      BD    0000G               JSR     CRTRST
    61          08BF      32                        PUL A
    62          08C0      97    00G                 STA A   ERRCD.D
    63          08C2      BD    0000G               JSR     IOCLNR
    64                    0001'                     .END
```

SYMBOL TABLE

ABRFLG=	0040	ABS	= 000000 G	ADRDEV=	000000 G	AFAIL	= 0030	AFPITT=	000000 G		
AMAT	= 0010	APPCOD=	000000 G	APPOLD	002CRG	ARRAY	= 0020	ASTR	= 0020		
ATLOD	= 0008	ATSNTG=	000000 G	ATVLD	= 0004	A.END	= 000000 G	A.MAX	= 000000 G		
A.PRIM=	000000 G	A.PTR	= 000000 G	A.SEC	= 000000 G	A.STAT=	000000 G	A.STRT=	000000 G		
A11X	= 000000 G	A14X	= 000000 G	A2X	= 000000 G	28X	= 000000 G	BAKABS=	000000 G		
BAKSTG=	000000 G	BANK	= 000000 G	BFRALC=	000000 G	BFRSTT=	0020	BLINK	= 000000 G		
BMAT	= 0004	BPIAMT=	000000 G	BRKCNT=	000000 G	BSTR	= 0008	BUFACT=	0010		
CDOPTR=	000000 G	CDSPTR=	000000 G	CHAR	= 000000 G	CHKDON	0470R	CNTCNT=	000000 G		
CLPTR	= 000000 G	CLRARG=	000000 G	CMAT	= 0001	COLCNT=	000000 G	CR	= 0000		
CRDC3	= 0002	CREOF	= 0008	CREOI	= 0004	CREOT	= 0020	CRETX	= 0010		
CRLF	= 000000 G	CRNORM=	0001	CRSTAT=	000000 G	CRTRST=	000000 G	CRVLD	= 0080		
CSTR	= 0002	CTKN	= 000000 G	CURSOR=	000000 G	DATDEV=	0022	DC3	= 0013		
DELY1S=	000000 G	DIMFLG=	0004	DIRCT	= 0080	DISBLE=	000000 G	DISCNT=	000000 G		
DISSRQ=	0008	DL	= 000000 G	DLTBLL=	000000 G	DP	= 000000 G	DRBUSY=	000000 G		
DREXTA=	000000 G	DREXTB=	000000 G	DSPDEV=	0020	DSPOUT=	000000 G	DSPSTT=	000000 G		
DT	= 000000 G	EDTBFR=	000000 G	EDTMAX=	000000 G	EDTPTR=	000000 G	ENDKEY=	0040		
EOFTYP=	0038	EOLCHR=	000000 G	EOLTG	= 000000 G	EOSTG	= 000000 G	ERRTSN=	000000 G		
ERDOWN=	000000 G	EREOM	= 000000 G	ERFBFR=	000000 G	ERFILE=	000000 G	ERFNFD=	000000 G		
ERIOE	= 000000 G	ERLOLD=	000000 G	ERMFMT=	000000 G	ERMILA=	000000 G	ERMOPN=	000000 G		
ERMTRD=	000000 G	ERNCRT=	000000 G	ERNIOD=	000000 G	ERNOLD=	000000 G	ERNSEP=	000000 G		
EROVRD=	000000 G	ERRCD	= 000000 G	ERTERM=	000000 G	ERUNDF=	000000 G	ERWRT	= 000000 G		
ESTG	= 000000 G	ETXCHR=	000000 G	EXTFLG=	0080	FILDEV=	0000	FILERQ=	000000 G		
FILES	= 000000 G	FILFID=	000000 G	FILINM	0701R	FILNAM	06F5R	FILLOC=	000000 G		
FILNAM	0709R	FIXNUM=	000000 G	FIX1	= 000000 G	FIX1A	= 0003	FIX1B	= 0004		
FMTVLD=	0008	FNDOK	028ER	FNFLG	= 0010	FORTG	= 000000 G	FPAITT=	000000 G		
FPC	= 000000 G	FRET	= 000000 G	GETLAR=	000000 G	GLAFLG=	000000 G	GOSTG	= 000000 G		
IDLC	= 000000 G	IMXTG	= 000000 G	INAITT=	000000 G	INITMT	064CRG	INPUTK=	0001		
INT	= 000000 G	IOBFR1=	000000 G	IOCLNR=	000000 G	IOFLGS=	000000 G	IOFUNC=	000000 G		
IOSCAN=	000000 G	ITM1TG=	000000 G	ITM2TG=	000000 G	ITTDEV=	0024	JMPX	= 000000 G		
KBDEV	= 001F	KBFLAG=	000000 G	KBIN	= 000000 G	KEYFLG=	0010	KEYSTK=	000000 G		
KIFILE	0029R	KILL	001ARG	KILTYP=	000000 G	LBRKTG=	000000 G	LCLFLG=	000000 G		
LDAX	= 000000 G	LDXX	= 000000 G	LENGTH=	000000 G	LISSOM	07C5R	LIST	= 000000 G		
LISTTG=	000000 G	LITCNT	0788R	LNNOTG=	000000 G	LOCTG	= 000000 G	LSP	= 000000 G		
LSTABT	0805R	LSTAGN	081AR	LSTEXT	07FFR	LSTFIX	07FFR	LSTFMT=	0002		
LSTFNC=	0017	LSTSND	082AR	LSXNRM	07F9R	MAGEND=	000000 G	MARKS	= 000000 G		
MKFILE	014ARG	MKFIL.	0160R	MKHDR	016CR	MODLOP	004CR	MODMTH	0042RG		
MODT88	003FRG	MRFIX	0126R	MRKDNN	01FDR	MRKFIL	01A3R	MRKIT	00F7R		
MRKITT	0186R	MRKOK	01E2R	MTAFIN	021DRG	MTASCI	02C2R	MTAO	0243R		
MTAOX	0242R	MTAOXX	023FR	MTBFR	= 000000 G	MTBGFL=	0002	MTBINT	0435RG		
MTBINY	028ER	MTBSWP	043ERG	MTCDON	0375R	MTCHKS	0461RG	MTCKIT	0336R		
MTCKSM=	0002	MTCLOS	0358RG	MTDATM	051DR	MTERR	= 000000 G	MTEXX	0487R		
MTEXXK	0536R	MTEXXQ	0527R	MTEXXR	0536R	MTEXX2	052AR	MTFDON	02D8R		
MTFFST=	0080 G	MTFIND	0211RG	MTFLEN=	0001	MTFLGS=	000000 G	MTFPOS=	0040 G		
MTFRD	0251R	MTFSIZ=	0080 G	MTFSRC	024AR	MTKEXX	008BR	MTKHDR	0083R		
MTKILL	00A4RG	MTKOK	00BCR	MTMARK	00E0RG	MTMAX	= 000000 G	MTMOK1	010CR		
MTM1	00F6R	MTM2	00F4R	MTM3	00F2R	MTNHDR=	0004	MTNULL	034CRG		
MTOPEN=	0020 G	MTPADR	0628RG	MTPDEV=	0021	MTPD2	= 0023	MTPIN	0575RG		
MTPINF	04EDR	MTPINR	0493RG	MTPINW	0471RG	MTPOLD	040BR	MTPOT	03F7R		
MTPOUT	03D8RG	MTPPRI	0501R	MTPRD	04F9R	MTPRG	02CER	MTPRGT	02C4R		
MTPRG.	02C6R	MTPRWD=	000000 G	MTPSAV	0537R	MTPSPC	0639R	MTPTR	= 000000 G		
MTPWRT	04BFR	MTP1	04F8R	MTRBFR	0583RG	MTRCON	033DR	MTRDER	0345RG		
MTRDON	034AR	MTREAD	02E8RG	MTRERX	04F4R	MTR.RD	02F0R	MTR.RR	02F7R		
MTSASC=	0008 G	MTSBIN=	0004 G	MTSCRT	0621RG	MTSECT	02D0R	MTSEX	0616R		
MTSEX1	0614R	MTSLST=	001C	MTSNEW=	0010 G	MTSPGM=	0002 G	MTSRD	= 0010 G		
MTSREG=	000000 G	MTSSEC=	0001 G	MTSSET	05FBRG	MTSTAT=	000000 G	MTSTT2=	000000 G		
MTSWRT=	0002 G	MTWAIT	0697R	MTWRIT	037ERG	MTWRT	0393R	MTXBYT=	000000 G		
MT.ERF	0282R	MT.I	0288R	MT.256	0268R	M.11	068ER	M.22	068FR		