```
                                        ; DIMSTR IS USED TO DIMENSION (ENTER A NEW STRING
                                        ; ENTRY INTO THE DATA TABLE.  ON ENTRY, X MUST
                                        ; POINT TO THE NAME TABLE ENTRY FOR THE STRING
                                        ; VARIABLE AND THE TWO BYTE INTEGER, INT1,
                                        ; MUST HAVE THE DIMENSION VALUE.
                                        ;
                                        ; DIMSTR DOES NOT CLEAR THE STACK, BUT
                                        ; DOES USE ALL OTHER HARDWARE REGISTERS AND R0 - R7.
                                        ;
                                        ; IF THERE IS NOT ENOUGH ROOM FOR THE ALLOCATION,
                                        ; AN ERROR IS INDICATED IN ERRCD.
                                        ;
                                        ;
018A  86  000G    DIMSTR: LDA A  RTRNTG,I   ; TAG RETURN.
018C  36                  PSH A
018D  DF  000G            STX    TPOINT,D        ; SAVE TABLE POINTER.
018F  BD  0000G           JSR    DISBLE    ; DISABLE BREAKS.
0192  BD  024A'           JSR    STGCAL    ; STORE INT1 AND BYTCNT.
0195  25  17              BCS    DSR       ; NO OVERFLOW, CONTINUE.
0197  BD  01FA'           JSR    ALOCAT    ; ENOUGH SPACE IN DATA TABLE?
019A  25  12              BCS    DSR       ; NO, ERROR EXIT.
019C  DE  000G            LDX    TPOINT,D  ; YES, GET TABLE ADDRESS.
019E  86  10              LDA A  20,I      ; GET DEFINED STRING ATTRIBUTE.
01A0  A7  04              STA A  4,X       ; STORE IN TABLE ENTRY.
01A2  4F                  CLR A            ; A = 0
01A3  A7  07              STA A  7,X       ; SET CURRENT LENGTH = 0.
01A5  A7  08              STA A  8.,X
01A7  BD  025F'           JSR    SETLSP    ; ALLOCATE SPACE IN DATA TABLE.
01AA  86  60              LDA A  140,I     ; GET UNDEFINED STRING VALUE ATTRIBUTE.
01AC  A7  02              STA A  2,X       ; STORE IN DATA HEADER.
01AE  BD  0C00G   DSR:    JSR    ENABLE    ; ENABLE BREAKS.
01B1  32                  PUL A            ; SCRATCH RETURN TAG.
01B2  39                  RTS              ; RETURN.
```

```
                                        ; GETVAL IS UES TO GET THE DIMENSION/SUBSCRIPT
                                        ; VALUES; IT MUST GET ONE VALUE, AND ALWAYS
                                        ; TRYS FOR TWO VALUES.  IF ONLY ONE VALUE IS SEEN, IT IS
                                        ; IN INT1; IF TWO VALUES HAVE BEEN CAPTURED, DIMCNT
                                        ; IS NON-ZERO AND THE ROW VALUE IS IN INT1 AND
                                        ; THE COLUMN VALUE IS IN INT2.  IF AN ERROR HAS
                                        ; OCCURED, CARRY AND ERRCD IS SET; A SUCCESSFUL
                                        ; RETURN HAS CARRY = 0.


01B3  BD  0000G     GETVAL: JSR   PSHRET    ; PUSH RETURN.
01B6  7F  0000G             CLR   DIMCNT    ; CLEAR DIMENSION COUNT(ASSUME 1)
01B9  9F  00G               STS   R0,D      ; SET UP R0 FOR TYPARG.
01BB  7F  0000G             CLR   R4        ; CLEAR TYPARG FLAGS.
01BE  8D  18                BSR   FIX       ; CONVERT FP VALUE TO INTEGER.
01C0  25  13                BCS   VALRET    ; FIX ERROR, RETURN.
01C2  32                    PUL A           ; GET NEXT ITEM TAG.
01C3  81  00G               CMP A LBRKTG,I  ; LEFT BRACKET?
01C5  27  0D                BEQ   B3        ; YES, FINISHED.
01C7  36                    PSH A           ; RESTORE TAG.
01C8  DE  00G               LDX   INT1,D    ; NO, BUBBLE UP VALUE
01CA  DF  00G               STX   INT2,D
01CC  8D  0A                BSR   FIX       ; GOT VALUE, MAKE INTEGER.
01CE  25  05                BCS   VALRET    ; FIX ERROR
01D0  73  0000G             COM   DIMCNT    ; SET INDICATOR = TWO VALUES.
01D3  32                    PUL A           ; SCRATCH LBRKTG.
01D4  0C            B3:     CLC             ; CLEAR ERROR INDICATOR.
01D5  7E  0000G     VALR' ': JMP  RTRN      ; RETURN.


                                        ; FIX IS USED TO CONVERT A FLOATING POINT NUMBER
                                        ; TO AN INTEGER.  IT LEAVES THE CONVERTED INTEGER IN
                                        ; INT1.  IF A FIX ERROR OCCURS, CARRY AND ERRCD IS SET ON EXIT
                                        ; OTHER WISE THEY ARE CLEAR ON EXIT.


01D8  BD  0000G     FIX:    JSR   PSHRET    ; PUSH RETURN.
01DB  BD  0000G             JSR   TYPARG    ; TRY TO GET VALUE.
01DE  26  06                BNE   FERR      ; NO VALUE THERE.
01E0  30                    TSX             ; SET UP INDEX REGISTER.
01E1  BD  0000G             JSR   FIX1      ; FIX NUMBER.
01E4  27  06                BEQ   FOK       ; NO ERRORS.
01E6  BD  02A1'     FERR:   JSR   ERROR     ; SET ERROR FLAG.
01E9  7E  0000G     FRET:   JMP   RTRN      ; RETURN.
01EC  EE  03        FOK:    LDX   3,X       ; GET INTEGER.
01EE  27  F6                BEQ   FERR      ; 0 VALUE = ERROR.
01F0  DF  00G               STX   INT1,D    ; SAVE IT.
01F2  86  09                LDA A 9.,I      ; GET STACK COUNT.
01F4  BD  0297'             JSR   CLEANS    ; CLEAN STACK.
01F7  0C                    CLC             ; CLEAR ERROR FLAG.
01F8  20  EF                BRA   FRET      ; RETURN.
```

```
                                          ; ALOCAT CHECKS TO SEE IF LSP+BYTCNT+FUDG
                                          ; WILL CAUSE A STACK COLLISION.  AS A LAST RESORT,
                                          ; COMPRESS IS CALLED, IF ALL FAILS, ERRCD IS
                                          ; SET = ERWSFL AND CARRY IS SET; OTHERWISE,
                                          ; CARRY = 0 AND THE ALLOCATION MAY PROCEED SAFETY.



01FA   86   00G       ALOCAT:  LDA A   RTRNTG,I      ; TAG RETURN.
01FC   36                      PSH A
01FD   96   00G                LDA A   BYTCNT,D      ; GET BYTE COUNT.
01FF   D6   01G                LDA B   BYTCNT+1,D
0201   CB   00G                ADD B   FUDGL,I       ; ADD IN FUDGE FACTOR.
0203   89   00G                ADC A   FUDGH,I
0205   25   22                 BCS     ALER          ; OVER 16 BITS WORTH.
0207   DB   01G                ADD B   LSP+1,D       ; ADD IN OLD LIMIT.
0209   99   00G                ADC A   LSP,D
020B   25   1C                 BCS     ALER          ; MORE THAN 16 BITS WORTH.
020D   9F   00G                STS     R7,D          ; TRANSFER UPPER LIMIT TO MEMORY.
020F   D0   01G                SUB B   R7+1,D        ; COMPARE TO LOWER LIMIT.
0211   92   00G                SBC A   R7,D
0213   25   0E                 BCS     ALRET         ; LIMITS OK, CONTINUE.
0215   2B   12                 BMI     ALER          ; OVER 32 K, COMPRESS CAN'T HANDLE.
0217   37                      PSH B                 ; PUSH IT TO STACK.
0218   36                      PSH A
0219   86   00G                LDA A   LTMITG,I      ; TAG IT.
021B   36                      PSH A
021C   BD   0000G              JSR     COMPR         ; TRY TO MAKE ROOM.
021F   96   00G                LDA A   ERRCD,D       ; SUCCESSFUL?
0221   26   03                 BNE     ABAD          ; NO, TAKE ERRO RET.
0223   0C             ALRET:   CLC                   ; C=0 FOR OK RETURN.
0224   32                      PUL A                 ; SCRATCH RETURN TAG.
0225   39                      RTS                   ; RETURN.
0226   0D             ABAD:    SEC                   ; C=1 FOR ERROR RETURN.
0227   32                      PUL A                 ; SCRATCH RETURN TAG.
0228   39                      RTS                   ; RETURN.
0229   86   00G       ALER:    LDA A   ERWSFL,I      ; GET OUT OF ROOM FLAG.
022B   97   00G                STA A   ERRCD,D       ; FLAG ERROR.
022D   20   F7                 BRA     ABAD          ; TAKE ERROR RETURN.
```

```
                                        ; SETDIM IS A COMMON ROUTINE USED BY SEVERAL
                                        ; SECTIONS OF ARRAY INITIAL DIMENSIONING TO TRANSFER
                                        ; THE USERS DIMENSIONS TO THE ARRAY NAME TABLE
                                        ; ENTRY.  THE INDEX REGISTER MUST BE POINTING
                                        ; TO THE FIST BYTE OF THE APPROPRIATE TABLE ENTRY.


022F  96  00G     SETDIM: LDA A  INT1,D       ; TRANSFER ROW VALUE TO TABLE.
0231  A7  05              STA A  5,X
0233  96  01G             LDA A  INT1+1,D
0235  A7  06              STA A  6,X
0237  96  00G             LDA A  DIMCNT,D     ; GET COLUMN INDICATOR.
0239  26  06              BNE    DOT2         ; GOT VALUE, TRANSFER
023B  4F                  CLR A               ; FORCE COLUMN VALUE TO 0.
023C  A7  07              STA A  7,X
023E  A7  08              STA A  8.,X
0240  39                  RTS                 ; RETURN.
0241  96  00G     DOT2:   LDA A  INT2,D       ; TRANSFER COLUMN VALUE TO TABLE.
0243  A7  07              STA A  7,X
0245  96  01G             LDA A  INT2+1,D
0247  A7  08              STA A  8.,X
0249  39                  RTS                 ; RETURN.
```

```
                                        ; STGCAL TRANSFERS INT1 TO THE TABLE ENTRY
                                        ; POINTED TO BY X AND STORES INT1+5 IN
                                        ; BYTCNT.  CARRY AND ERRCD IS SET ON RETURN IF AN OVERFLOW
                                        ; HAS OCCURED.


024A  96  00G     STGCAL: LDA A  INT1,D          ; GET DIMENSION COUNT
024C  D6  01G             LDA B  INT1+1,D
024E  A7  05              STA A  5,X           ; STORE IN TABLE ENTRY.
0250  E7  06              STA B  6,X
0252  CB  05              ADD B  5,I           ; ADD IN HEADER VALUE.
0254  89  00              ADC A  0,I
0256  97  00G             STA A  BYTCNT,D      ; SAVE TOTAL BYTE COUNT.
0258  D7  01G             STA B  BYTCNT+1,D
025A  24  02              BCC    STGR          ; NO OVERFLOW, RETURN.
025C  8D  43              BSR    ERROR         ; INDICATE OVERFLOW.
025E  39          STGR:   RTS                  ; RETURN.


                                        ; SETLSP IS USED TO STORE THE CURRENT VALUE OF LSP
                                        ; IN THE NAME TABLE AS THE DATA POINTER AND
                                        ; CALCUALTES/SETS UP A NEW VALUE OF LSP
                                        ; ACCORDING TO THE EQUATION:

                                        ; LSP = LSP+BYTCNT.

                                        ; ON EXIT, X IS POINTING
                                        ; TO THE NEW DATA ENTRY (OLD VALUE OF LSP).


025F  96  00G     SETLSP: LDA A  LSP,D          ; SET UP DATA POINTER.
0261  A7  0B              STA A  11.,X
0263  D6  01G             LDA B  LSP+1,D
0265  E7  0C              STA B  12.,X
0267  DB  01G             ADD B  BYTCNT+1,D    ; CALCUALTE NEW LSP.
0269  99  00G             ADC A  BYTCNT,D
026B  DE  00G             LDX    LSP,D         ; GET NEW DATA ENTRY ADDRESS.
026D  97  00G             STA A  LSP,D         ; SET LSP TO NEW VALUE.
026F  D7  01G             STA B  LSP+1,D
0271  96  00G             LDA A  BYTCNT,D      ; GET BYTE COUNT.
0273  D6  01G             LDA B  BYTCNT+1,D
0275  A7  00              STA A  0,X           ; STORE IN DATA HEADER.
0277  E7  01              STA B  1,X
0279  39                 RTS                  ; RETURN.
```

```
                                        ; BYTCAL PERFORMS THE FOLLOWING CALCUALTION:
                                        ;
                                        ;   ((A8+(INT1*INT2))*8)+5
                                        ;
                                        ; CARRY IS SET TO INDICATE OVERFLOW (OVER 16 BITS).
                                        ; OTHERWISE IT IS 0.
                                        ;
027A DE  00G      BYTCAL: LDX   INT1,D      ; GET OPERAND 1.
027C DF  00G              STX   R7,D        ; SET UP FOR MULTIPL.
027E DE  00G              LDX   INT2,D      ; GET OPERAND 2.
0280 8D  2F               BSR   INTMLR      ; MULTIPLY THEM.
0282 26  10               BNE   CALERR      ; IF HIGH HALF SET, TOO BIG.
0284 DE  00G              LDX   R7,D        ; GET INTEGER.
0286 DF  00G              STX   DIMLP,D     ; SET UP DIMENSION LOOP COUNT.
0288 CE  0008             LDX   #8.*1       ; GET FP NUMBER SIZE.
028B 4F                   CLR A             ; GET HEADER OFFSET.
028C C6  05               LDA B 5,I
028E 8D  21               BSR   INTMLA      ; EXPLODE ELEMENT STORAGE.
0290 26  02               BNE   CALERR      ; NOT ENOUGH ROOM.
0292 0C                   CLC               ; CLEAR INTEGER OVERFLOW FLAG.
0293 39                   RTS               ; RETURN.
0294 8D  08       CALERR: BSR   ERROR       ; SET OVERFLOW ERROR.
0296 39                   RTS               ; RETURN.
                                        ;
                                        ; CLEANS IS USED TO REMOVE (CLEAN) THE NUMBER
                                        ; OF BYTES OFF THE HARDWARE STACK AS INDICATED BY
                                        ; THE LOOP COUNT IN 'B'.  NO ERRORS ARE POSSIBLE FROM
                                        ; CLEANS.
                                        ;
0297 8D  0000G    CLEANS: JSR   PSHRET      ; PUSH RETURN.
029A 33           CLP:    PUL B             ; LOOP UNTIL STACK CLEAN.
029B 4A                   DEC A
029C 26  FC               BNE   CLP
029E 7E  0000G            JMP   RTRN        ; RETURN.
```

```
                                          ; ERROR IS THE COMMON SET ERROR ROUTINE.
                                          ; IT HANDLES ALL ERRORS EXCEPT "NO SPACE".


02A1   96   00G        ERROR:  LDA A   DSFLG,D     ; GET SUB/DIM POINTER/FLAG.
02A3   26   06                 BNE     SUBR        ; WE ARE IN SUBSCRIPTING.
02A5   86   00G                LDA A   ERDIM,I     ; GET DIMENSION ERROR.
02A7   97   00G        ERFIN:  STA A   ERRCD,D     ; FLAG IT.
02A9   0D                      SEC                 ; SET INTERNAL ERROR FLAG.
02AA   39                      RTS                 ; RETURN.
02AB   86   00G        SUBR:   LDA A   ERSUBC,I    ; GET SUBSCRIPT ERROR.
02AD   20   F8                 BRA     ERFIN       ; FINISH.
```

```
                                        ; INTMLA/C IS USED TO DO A 16 BIT BY 16 BIT
                                        ; UNSIGNED INTEGER MULTIPLY TO PRODUCE A 32
                                        ; BIT UNSIGNED PRODUCT; INTEGER MULTIPLY HAS
                                        ; TWO ENTRY POINTS:

                                        ; INTMLC DOES R6, R7 <- X*R7

                                        ; INTMLA DOES R6, R7 <- (X*R7)+AB

                                        ; WITH R6 HIGH BYTE AND R7+1 LOW BYTE.


02AF  4F              INTMLC: CLR A             ; CLEAR REMAINDER.
02B0  5F                      CLR B
02B1  DF    00G       INTMLA: STX   R6,D        ; SAVE OPERAND 1.
02B3  CE    0010              LDX   16.,I       ; GET LOOP COUNT.
02B6  76    0000G             ROR   R7          ; ROTATE OPERAND 2.
02B9  76    0001G             ROR   R7+1
02BC  24    04        1$:     BCC   2$          ; NO CARRY, DO NO ADD.
02BE  DB    01G               ADD B R6+1,D      ; ADD IN PARTIAL PRODUCT.
02C0  99    00G               ADC A R6,D
02C2  46              2$:     ROR A             ; ROTATE PARTIAL PRODUCT.
02C3  56                      ROR B
02C4  76    0000G             ROR   R7
02C7  76    0001G             ROR   R7+1
02CA  09                      DEX               ; BUMP DOWN LOOP COUNT.
02CB  26    EF                BNE   1$          ; CONTINUE LOOP.
02CD  97    00G               STA A R6,D        ; AB = HIGH ORDER HALF.
02CF  D7    01G               STA B R6+1,D
02D1  DE    00G               LDX   R6,D        ; X = HIGH ORDER HALF.
02D3  39                      RTS               ; RETURN.
```

```
    1                0001'                       .END
```

```
DIMSUB  DIMENSION/SUBSCRIPTING        RT-11 MMAC  VM02-10   14-OCT-76 01:30:32 PAGE 15+
SYMBOL TABLE

ABAD    0226R       ALER    0229R       ALOCAT  01FAR       ALRET   0223R       A8X   = XXXXXX G
BRET    0178R       BRKCNT= XXXXXX G    BYTCAL  027ARG      BYTCNT= XXXXXX G    B11     0178R
B12     0150R       B3      01D4R       B6      00B0R       B7      00F5R       CA      00CBR
CALERR  0294R       CBRET   0175R       CLEANS  0297R       CLP     029AR       COMPR = XXXXXX G
DIM     0000RG      DIMCAL  00C2R       DIMCNT= XXXXXX G    DIMERR  00B0R       DIMFLG= 0004
DIMLP = XXXXXX G    DIMS    013FR       DIMSON  00A9R       DIMSTR  018ARG      DIMSUB= 0000RG
DISBLE= XXXXXX G    DMER    0185R       DOALC   00FAR       DOT2    0241R       DR      00F2R
DRFIN   013AR       DSFLG = XXXXXX G    DSR     01AER       ENABLE= XXXXXX G    ERDIM = XXXXXX G
ERFIN   02A7R       ERRCD = XXXXXX G    ERROR   02A1R       ERSUBC= XXXXXX G    ER4SFL= XXXXXX G
FERR    01E6R       FIX     01D8R       FIX1  = XXXXXX G    FOK     01ECR       FRET    01E9R
FUDGH = XXXXXX G    FUDGL = XXXXXX G    GETVAL  0183R       GOTV    002CR       INIMLA  0281RG
INTMLC  02AFRG      INT1  = XXXXXX G    INT2  = XXXXXX G    ITM1TG= XXXXXX G    LBRKTG= XXXXXX G
LCLFLG= XXXXXX G    LP      0128R       LSP   = XXXXXX G    PAETG = XXXXXX G    PNTNTG= XXXXXX G
PNTSTG= XXXXXX G    PSCTG = XXXXXX G    PSHPET= XXXXXX G    RPN     0000RG      RPNCK   0018R
RPNFIN  0024R       RRET    0040R       RTRN  = XXXXXX G    RTRNTG= XXXXXX G    R0    = XXXXXX G
R1    = XXXXXX G    R2    = XXXXXX G    R3    = XXXXXX G    R4    = XXXXXX G    R5    = XXXXXX G
R6    = XXXXXX G    R7    = XXXXXX G    SETDIM  022FR       SETLSP  025FR       SETS    0015R
STGCAL  024AR       STGR    025ER       SUBCAL  0043R       SUBERR  003DR       SUBFIN  0082R
SUBR    02ABR       SUB2    0063R       TPOINT= XXXXXX G    TYPARG= XXXXXX G    VALRET  01D5R
VALTG = XXXXXX G

.  ABS.    0000    00
           02D4    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3088. WORDS
,SY:DIMSUB/C<DK1:SEICLI,DIMSUB
```

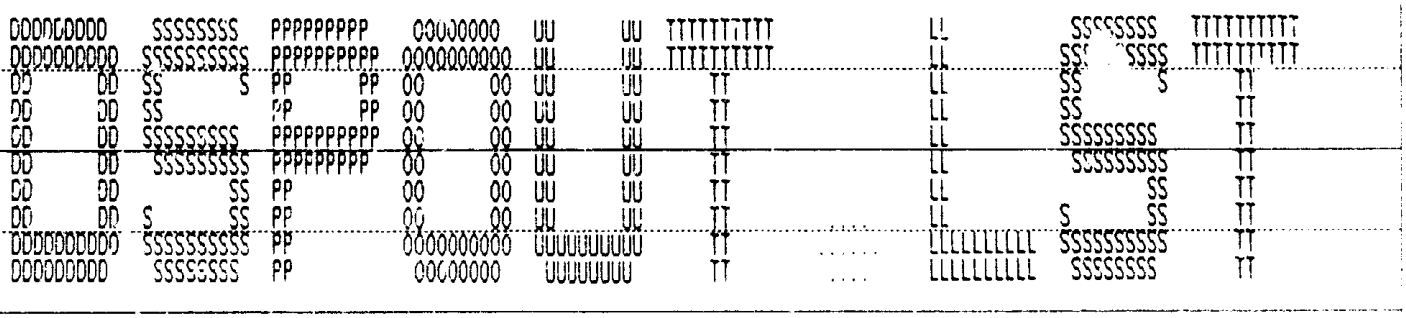
CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

	11-9	12-10												
INT2	1-37#	4-27*	4-36	4-37	4-39	4-41*	4-43	4-44	4-46*	5-22*	8-24*	10-19	10-21	12-12
INTMLA	1-23#	12-13	12-20	14-16#										
INTMLC	1-26#	14-14#												
ITM1TG	1-71#	9-26												
LBRKTG	1-70#	8-20												
LCLFLG	1-39#	2-10	2-11*	3-16										
LP	5-72#	5-80												
LSP	1-57#	5-52	5-53	9-16	9-17	11-34	11-36	11-40	11-41*	11-42*				
PAETG	1-65#	4-70												
PNTNTG	1-66#													
PNTSTG	1-68#	5-8												
PSCTG	1-67#													
PSHRET	1-32#	3-9	8-13	8-41	12-36									
R0	1-41#	8-15*												
R1	1-42#													
R2	1-43#													
R3	1-44#													
R4	1-45#	8-16*												
R5	1-46#													
R6	1-47#	14-16*	14-21	14-22	14-29*	14-30*	14-31							
R7	1-48#	4-62	4-63	5-27	5-54*	5-55*	5-70	5-77	5-79*	9-19*	9-20	9-21	12-11*	12-15
	14-18*	14-19*	14-25*	14-26*										
RPN	1-20#	3-9#												
RPNCK	3-12	3-16#												
RPNFIN	3-15	3-20#												
RRET	4-12#	4-72												
RTRN	1-33#	3-22	4-12	5-45	6-40	8-30	8-48	12-40						
RTRNTG	1-69#	7-15	9-9											
SETDIM	5-41	5-61	10-9#											
SETLSP	5-66	7-29	11-34#											
SETS	3-13#	3-18												
STGCAL	7-19	11-8#												
STGR	11-16	11-18#												
SUB2	4-23	4-31#												
SUBCAL	4-10	4-13#												
SUBERR	4-11#	4-17	4-25	4-32	4-38									
SUBFIN	4-30	4-47*												
SUBR	13-6	13-11#												
TPOINT	1-54#	3-26*	4-21	4-33	4-42	4-55	4-56	4-59	5-13	5-29	5-58	6-15	6-33	7-17*
	7-23													
TYPARG	1-21#	8-42												
VALRET	8-18	8-26	8-30#											
VALTG	1-64#													

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-3# 5-40 5-56



DSPOUT DISPLAY OUTPUT CONTROL RT-11 MMAC VM02-10 14-OCT-76 01:30:48
TABLE OF CONTENTS

```
                        .TITLE  DSPOUT DISPLAY OUTPUT CONTROL
                        .IDENT  /S800372/
                        .GLOBL  DSPCHR          ; OUTPUT CHARACTER TO SCREEN (CNTL IS
                        .GLOBL  BANKSW          ; BANK SWITCH ADDRESS
                                                ; DISPLAYED AS UNDERLINED CHARACTERS)
                        .GLOBL  CTLCHR          ; OUTPUT CHARACTER DIRECTLY
                        .GLOBL  FULSCN          ; SCANS THE "FULL" MESSAGE
                        .GLOBL  DSPOUT          ; OUTPUTS THE BUFFER TO THE SCREEN
                        .GLOBL  PCHAR,PCHARZ,FULL,BLINK
```

```
                                                .SBTTL  SEND ANY CHARACTER TO THE DISPLAY
                                        ; CTLCHR -- THIS ROUTINE SENDS THE CHARACTER IN
                                        ; ACCUMULATOR A AS IS, CONTROL CHARACTER
                                        ; OR NOT.  IF THE DISPLAY IS FULL THIS
                                        ; ROUTINE WILL WAIT UNTIL THE PAGE HAS
                                        ; BEEN CLEARED (NOTE:  IT WILL NOT PUSH
                                        ; SPECIAL STATUS FOR PUTCHR).
                                                .GLOBL  DSPVCT
                                                .GLOBL  ONTBL,BAKSTG,XFNBNK
0000    F6  0002G       CTLCHR: LDA B   DSPVCT+2        ; TERMINAL MODE ACTIVE?
0003    27  0B                  BEQ     1$
0005    36                      PSH A
0006    17                      TBA                     ; SWITCH TO TERMINAL CONTROLLER
0007    BD  0000G               JSR     SETBNK
                                                .GLOBL  SETBNK,BNKADR
000A    32                      PUL A                   ; GET CHARACTER BACK
000B    FE  0000G               LDX     DSPVCT
000E    AD  00                  JSR     0,X
0010    36              1$:     PSH A                   ; SAVE CHARACTER FOR AFTER PAGE
0011    BD  0000G               JSR     PCHAR           ; PRINT THE CHARACTER
0014    7D  0000G               TST     FULL            ; DID THIS MAKE PAGE FULL
0017    27  5D                  BEQ     RTN             ; RETURN IF ALL OK
                                        ; PAGE FULL LOGIC
                                        ; DECIDE WHETHER TO FLASH "F" OR DO ON FUL
                                        ;
0019    96  00G                 LDA A   KBFLAG,D        ; KEYBOARD OPEN?
001B    88  02                  EOR A   RUNN,I
001D    85  03                  BIT A   INPUTK+RUNN,I
001F    26  4C                  BNE     FLASH
                                                .GLOBL  FULSTT
                                        ;
                                        ; 0 = NORMAL
                                        ; 1 = HOME
                                        ; 2 = PAGE
                                        ; 3 = COPY & PAGE
                                        ;
0021    B6  0000G               LDA A   FULSTT          ; GET PAGE FULL FUNCTION
0024    27  14                  BEQ     OTHFUL
0026    4A                      DEC A
0027    27  0A                  BEQ     FULHOM
0029    4A                      DEC A
002A    27  03                  BEQ     FULPAG
002C    BD  011D'               JSR     DSPCPY          ; COPY AND PAGE
002F    86  0C          FULPAG: LDA A   14,I            ; PAGE
0031    20  02                  BRA     FULFNC
0033    86  1E          FULHOM: LDA A   36,I            ; HOME
0035    BD  0000G       FULFNC: JSR     PCHAR
0038    20  36                  BRA     FULDON
003A    FE  0004G       OTHFUL: LDX     ONTBL+4         ; IS ON UNIT ACTIVE?
003D    27  26                  BEQ     FLASH           ; NOPE
003F    96  00G                 LDA A   10FUNC,D        ; IF GRAPHICS THEN NO FULL
0041    84  3F                  AND A   63.,I           ; TURN OFF HIGH BITS
0043    81  14                  CMP A   20.,I
0045    27  26                  BEQ     FLASH
0047    81  15                  CMP A   21.,I
0049    27  22                  BEQ     FLASH
004B    86  00G                 LDA A   BAKSTG,I        ; IT'S FOR REAL
```

SEND ANY CHARACTER TO THE DISPLAY

```
0040   36                        PSH A
004E   B6   0003G                LDA A   XFNBNK+3          ; IS ANY BODY HOME TO HELP WITH PAGE FULL?
0051   27     17                 BEQ     2$
0053   D6    00G                 LDA B   BANK,D
0055   37                        PSH B
0056   37                        PSH B
0057   C6    00G                 LDA B   ITM1TG,I
0059   37                        PSH B
005A   BD   0000G                JSR     SETBNK
005D   FE   000AG                LDX     BNKADR+10     ; GET PROPER VECTOR
0060   EE     00                 LDX     0,X
0062   AD     00                 JSR     0,X           ; GO DO FULL RECURSSIONS
0064   31                        INS                   ; CLEAN STACK
0065   31                        INS
0066   32                        PUL A                 ; SET UP OLD BANK
0067   BD   0000G                JSR     SETBNK
006A   31               2$:      INS                   ; REMOVE TAG
006B   20     03                 BRA     FULDON        ; GO CLEAN UP
006D   BD   0094'       FLASH:   JSR     FULSCN        ; FLASH THE "F"
0070   32               FULDON:  PUL A                 ; RETRY THE CHARACTER
0071   81     0D                 CMP A    #0D,I  ; IF "CR" THEN DON'T RETRY
0073   26     88                 BNE     CTLCHR
0075   39                        RTS                   ; RETURN
0076   32               RTN:     PUL A
0077   39                        RTS
```

```
                                        .SBTTL  DSPCHR OUTPUT ASCII ONLY
                                ; DSPCHR -- THIS ROUTINE PRINTS CHARACTERS
                                ; TO THE DISPLAY BY USING CTLCHR.  IT HOWEVER
                                ; CONVERTS ALL CONTROL CHARACTERS TO
                                ; ALPHA CHAR., BACK SPACE, UNDERLINE.
0078    C6      006     DSPCHR: LDA B   RTRNTG,I        ; TAG THE STACK
007A    37                      PSH B
007B    85      E0              BIT A   #0E0,I  ; TEST FOR CONTROL CHARACTERS
007D    26      10              BNE     PNTIT           ; IF NOT CONTROL PRINT NORMAL
007F    81      0D              CMP A   #0D,I   ; IF IT IS CR THEN PRINT IT
0081    27      0C              BEQ     PNTIT
0083    8A      40              ORA A   #40,I   ; MAKE IT ALPHA
0085    BD      0000'           JSR     CTLCHR          ; PRINT IT
0088    86      08              LDA A   #8.,I           ; PRINT BACK SPACE
008A    BD      0000'           JSR     CTLCHR
008D    86      5F              LDA A   #5F,I   ; PRINT UNDERLINE
008F    BD      0000'   PNTIT:  JSR     CTLCHR
0092    31                      INS                     ; REMOVE TAG
0093    39                      RTS
```

```
                                        .SBTTL  FULSCN SCAN THE "FULL" MESSAGE
                                ; FULSCN -- THIS ROUTINE DOES THE REFRESHER
                                ; OF "FULL" ON THE TUBE WHEN IT IS
                                ; WAITING FOR A PAGE COMMAND.
                                        .GLOBL  CURSOR          ; THE CURSOR LOCATION
                                        .GLOBL  TMPHY           ; Y VECTOR
                                        .GLOBL  TMPHX           ; X VECTOR
                                        .GLOBL  VECTOR          ; PUTS VECTORS TO THE SCREEN
                                        .GLOBL  TMP1
                                ;
0094    96      00G     FULSCN: LDA A   CURSOR.D        ; SAVE PRESENT CURSOR
0096    36                      PSH A
0097    86      46              LDA A   'F,I            ; GET AN "F"
0099    97      00G             STA A   CURSOR.D
009B    7C      0000G           INC     TMP1            ; MAKE IT MOVE TO CORNER
009E    CE      0000            LDX     0.,I            ; EXTABLISH X,Y
00A1    DF      00G             STX     TMPHX.D
00A3    CE      0320            LDX     800.,I
00A6    DF      00G             STX     TMPHY,D
00A8    BD      0000G           JSR     VECTOR          ; GO DO IT TO SCREEN
00AB    BD      0141'   SCAN:   JSR     GENCUR          ; DO A SCAN
00AE    96      00G             LDA A   FULL,D          ; WAITING FOR CLEAR PAGE
00B0    26      F9              BNE     SCAN
00B2    32                      PUL A                   ; RESET OLD CURSOR
00B3    97      00G             STA A   CURSOR,D
00B5    7F      0000G           CLR     TMP1            ; RESET DISPLAY
00B8    7F      0000G           CLR     BLINK           ; RESET BLINK MODE
00BB    39                      RTS                     ; RETURN
```

```
                                        .SBTTL  DSPOUT OUTPUT THE OUTPUT BUFFER TO THE SCREEN
                                ;THIS ROUTINE PUTS THE OUTPUT BUFFER TO THE SCREEN (ASCII)
                                ;       A.END---IS POINTER TO LAST VALID CHARACTER
                                ;       A.PTR---IS POINTER TO THE FIRST VALID CHARACTER
                                ;
                                        .GLOBL  NOOUT
                0002                    LSTFMT  =  #002         ; LIST FORMAT REQUESTED
                                ;
008C  86  000   DSPOUT: LDA A   RTRNTG,I        ; TAGIT
008E  36                PSH A
008F  96  000           LDA A   A.SEC.D         ; TEST FOR VALID SEC. ADDR.
00C1  81  09            CMP A   9.,I            ; DIRECTORY?
00C3  27  29            BEQ     DSPGO
00C5  81  0C            CMP A   12.,I           ; PRINT?
00C7  27  25            BEQ     DSPGO
00C9  81  13            CMP A   19.,I           ; LIST?
00CB  27  21            BEQ     DSPGO
00CD  81  16            CMP A   22.,I           ; PAGE?
00CF  27  0E            BEQ     DSPPGE
00D1  81  17            CMP A   23.,I           ; HOME?
00D3  27  0E            BEQ     DSPHOM
00D5  81  0A            CMP A   10.,I           ; COPY?
00D7  27  27            BEQ     DSCOPY
00D9  81  18            CMP A   24.,I           ; REFRESH CRC
00DB  27  28            BEQ     DSRFR
00DD  31                INS
00DE  39                RTS
00DF  86  0C    DSPPGE: LDA A   14,I             ; GET PAGE
00E1  20  02            BRA     DSPFNC
00E3  86  1E    DSPHOM: LDA A   36,I             ; GET HOME
00E5  BD  0000G DSPFNC: JSR     PCHAR
00E8  86  10    DSPEND: LDA A   #10.,I          ; SET END I/O
00EA  97  000           STA A   IOFLGS,D
00EC  31        DSPDON: INS
00ED  39                RTS                     ; RETURN
00EE  DE  00G   DSPGO:  LDX     A.PTR,D         ; GET POINTER TO NEXT CHARACTER
00F0  A6  00    DSPL:   LDA A   0,X             ; GET CHARACTER
00F2  BD  0000' 1$:     JSR     CTLCHR          ; OUT PUTIT
00F5  DE  00G           LDX     A.PTR,D         ; GET POINTER AGAIN
00F7  9C  00G           CPX     A.END,D         ; DONE YET?
00F9  27  F1            BEQ     DSPDON
00FB  08                INX
00FC  DF  00G           STX     A.PTR,D         ; STORE POINTER TO NEXT CHARACTER
00FE  20  F0            BRA     DSPL            ; AROUND WE GO
                ;*****************************************************************
                ;       DISPLAY SPECIAL FEATURE CREATURES
                ;
010C  BD  011D' DSCOPY: JSR     DSPCPY
010F  20  E3            BRA     DSPEND
                ;
                ;
0111  DE  00G   DSRFR:  LDX     A.PTR,D         ; DISPLAY REFRESHED CHAR. FOR
0113  A6  00            LDA A   0,X             ; APPROX. 1/4 SEC.
0115  97  00G           STA A   CURSOR,D        ; SET IT
0117  73  0000G         COM     BLINK           ; SET WRITE-THRU
011A  86  15            LDA A   #NTIM,I
011C  36        1$:     PSH A
```

```
0111    BD   0146'          JSR    DOIT        ; REFRESH
0114    32                  PUL A
0115    4A                  DEC A
0116    26     F8           BNE    1$          ; DONE?
0118    7F   0000G          CLR    BLINK
011B    20     CB           BRA    DSPEND
```

```
                                                .SBTTL  DSPCPY INITIATES A HARD COPY OF THE SCREEN
                                                .GLOBL  DSPCPY
                        0020                    HRDCPY  =  H20
                                                ;THIS ROUTINE TESTS IF S'ILL DOING A VECTOR OR ANOTHER SCREEN FUNCTION
                                                ;       BEFORE INITIATING A HARD COPY
                                                .GLOBL  PIAHX           ; THE STATUS HALF
                                                .GLOBL  WAIT.
                                                .GLOBL  DRBUSY
                                                .GLOBL  DSPSTT,CPYFLG,DSPCP2
                                                .GLOBL  PIALX
                                                ;
011D    96      00G     DSPCPY: LDA A   DSPSTT,0        ; SEE IF DISPLAY DISABLED
011F    2A      08              BPL     DSRLCP
0121    8A      00G             ORA A   CPYFLG,1        ; TURN ON COPY LATER BIT
0123    97      00G             STA A   DSPSTT,0
0125    39                      RTS
0126    BD      012C'   DSPCP2: JSR     DSRLCP          ; MAKE COPY
0129    7E      0000G           JMP     DRBUSY          ; AND WAIT
                                ;
012C                    DSRLCP:
012C    C6      3C              LDA B    HO3C,1
012E    F7      0001G           STA B   PIAHX+1
0131    F6      0000G           LDA B   PIAHX           ; GET DISPLAY STATUS
0134    C4      DF              AND B   -1-HRDCPY,1     ; DIDDLE HARD COPY LINE
0136    F7      0000G           STA B   PIAHX
0139    CA      20              ORA B   HRDCPY,1
013B    F7      0000G           STA B   PIAHX
013E    7E      0000G           JMP     WAIT.           ; MAKE SURE DRBUSY HAS TIME TO RISE
```

```
                                        .SBTTL  BLINKING CURSOR GENERATOR
                                        .GLOBL  GENCUR
                                        .GLOBL  PGCTR           ; PAGE SREEN COUNTER
                                        .GLOBL  PCHAR
                                ;EACH ENTRY WILL DECREMENT A COUNTER AND WILL TOGGLE
                                ;       WRITE-THRU MODE APPROPRIATELY
                                ;IT ASSUMES THAT CURSOR HAS BEEN SETUP.
                        0090            OFFTIM  =  H090         ; CURSOR OFF TIMER
                        0015            ONTIM   =  H15          ; CURSOR ON TIMER
0141    7A      0000G   GENCUR: DEC     DISCNT          ; CURSOR BLINK COUNTER
0144    27      11              BEQ     NEWMOD          ; TOGGLE MODES IF ZERO
0146    B6      0000G   DOIT:   LDA A   BLINK           ; WHICH STATE?
0149    27      06              BEQ     WAIT1
0148    96      00G             LDA A   CURSOR,D        ; GET CURSOR CHARACTER AND BLINK
014D    BD      0000G           JSR     PCHAR           ;       CONTROLS WRITE-THRU
0150    39                      RTS
0151    86      55      WAIT1:  LDA A   #055.!  ; WAIT1 COUNTER
0153    4A      1$:     DEC A
0154    26      FD              BNE     1$              ; DONE?
0156    39                      RTS
0157    73      0000G   NEWMOD: COM     BLINK           ; TOGGLE STATE
015A    26      07              BNE     WTHRU           ; NOW WHICH STATE?
015C    86      90              LDA A   OFFTIM,!        ; SET UP FOR OFF TIME
015E    B7      0000G           STA A   DISCNT
0161    20      E3              BRA     DOIT            ; GO DO IT
0163    FE      0000G   WTHRU:  LDX     PGCTR           ; GET PAGE COUNTER
0166    27      0B              BEQ     WTHRU1          ; IF ALLREADY PAGED THEN LEAVE
0168    09                      DEX
0169    FF      0000G           STX     PGCTR           ; DEC. COUNT
016C    26      05              BNE     WTHRU1          ; IF NOT ZERO YET DON'T PAGE
016E    86      0C              LDA A   14,!            ; ELSE PAGE THE SCREEN
0170    BD      0000G           JSR     PCHAR
0173    86      15      WTHRU1: LDA A   ONTIM,!         ; SET UP FOR ON TIME
0175    B7      0000G           STA A   DISCNT
0178    20      CC              BRA     DOIT            ; GO DO IT
```

```
                                        .SBTTL  DSGRAF--DISPLAY MOVE AND DRAW ALGORITHM
                                ;THESE TWO ROUTINES CONVERT A FLOATING NUMBER TO AN INTEGER AND
                                ;SIMULTANIOUSLY CONVERTS FROM GDU'S TO TEKPOINTS.  THANKS TO FLOATING
                                ;POINT JOHN.
                                        .GLOBL  DSMOVE          ; MOVE ENTRY PONT
                                        .GLOBL  DSDRAW          ; DRAW ENTRY POINT
                                        .GLOBL  DSGRAF          ; GENERAL PURPOSE ENTRY POINT
                                        .GLOBL  FIX1            ; FIXES NUMBER
                                        .GLOBL  TMPHX           ; GRAPHICS STORAGE
                                        .GLOBL  TMPLX
                                        .GLOBL  TMPHY
                                        .GLOBL  TMPLY
                                        .GLOBL  TMP1            ; MOVE/DRAW INDICATOR
                                        .GLOBL  TMP2            ; FLAG
                                        .GLOBL  YAXIS           ; AXIS FLIP-FLOP
                                        .GLOBL  VECTOR          ; VECTOR ROUTINE FOR DISPLY
                                        .GLOBL  POINT           ; LOCATION OF FLOATING CONSTANT
                                        .GLOBL  PSHFPN          ; PUSH FLOATING NUMBER ON STACK
                                        ;       SOME NUMBERS DO FLOAT (DON'T THEY?)
                                        .GLOBL  A9X
                                        .GLOBL  GSCNUM,FPMUL
                                ;
017A    86      01      DSMOVE: LDA A   1,I             ; SET MOVE
017C    97      00G             STA A   TMP1,D
017E    20      03              BRA     DSGRAF
0180    7F      0000G   DSDRAW: CLR     TMP1            ; SET DRAW
0183    DE      00G     DSGRAF: LDX     POINT,D         ; PUT GDU INFO ON STACK
0185    BD      0000G           JSR     PSHFPN
0188    30                      TSX
0189    60      01              TST     1,X
018B    2B      50              BMI     DSGERR          ; TEST FOR NEG. NUMBER
018D    E6      02              LDA B   2,X
018F    A6      01              LDA A   1,X
0191    CB      07              ADD B   7,I
0193    89      00              ADC A   0,I
0195    84      07              AND A   7,I
0197    A7      01              STA A   1,X
0199    E7      02              STA B   2,X
019B    BD      0000G           JSR     FIX1
019E    26      4A              BNE     DSGERR
01A0    A6      03              LDA A   3,X
01A2    E6      04              LDA B   4,X
01A4    44                      LSR A
!!!!!
01A5    56                      ROR B
01A6    44                      LSR A
01A7    56                      ROR B
01A8    E0      03              SUB B   3,X
01AA    82      00              SBC A   0,I
01AC    64      03              LSR     3,X
01AE    E0      03              SUB B   3,X
01B0    82      00              SBC A   0,I
01B2    44                      LSR A
01B3    56                      ROR B
01B4    44                      LSR A
01B5    56                      ROR B
01B6    A7      03              STA A   3,X
01B8    E7      04              STA B   4,X
```

```
        01BA    85  FC              BIT A   252.,I          ; TEST FOR OVERFLOW
        01BC    26  2C              BNE     DSGERR
        01BE    EE  03              LDX     FIX1A,X
        01C0    7D  0000G           TST     YAXIS           ; TEST WHICH AXIS
        01C3    2B  06              BMI     DSDOIT
        01C5    DF  00G             STX     TMP2,D
        01C7    86  80              LDA A   128.,I
        01C9    20  17              BRA     DSGDON
        01CB    88  03      DSDOIT: EOR A   3,I             ; FURTHER TEST FOR YAXIS
        01CD    26  04              BNE     1$
        01CF    C1  13              CMP B   19.,I           ; OVERFLOW
        01D1    22  17              BHI     DSGERR
        01D3    96  00G     1$:     LDA A   YAXIS,D
        01D5    47                  ASR A                   ; TEST IF HAVE ERROR
        01D6    25  09              BCS     DSGEXT
        01D8    DF  00G             STX     TMPHY,D         ; SET UP YAXIS
        01DA    DE  00G             LDX     TMP2,D          ; DRAW VECTOR
        01DC    DF  00G             STX     TMPHX,D
        01DE    BD  0000G           JSR     VECTOR
        01E1    4F          DSGEXT: CLR A
        01E2    97  00G     DSGDON: STA A   YAXIS,D
        01E4    30                  TSX
        01E5    BD  0000G           JSR     A9X
        01E8    35                  TXS                     ; CLEAN UP STACK
        01E9    39                  RTS
        01EA    7F  0000G   DSGERR: CLR     ERRCD
        01ED    96  00G             LDA A   YAXIS,D
        01EF    2B  F0              BMI     DSGEXT
        01F1    86  81              LDA A   129.,I
        01F3    20  ED              BRA     DSGDON
                            ;
                            ;UNIVERSAL GRAPHIC CONSTANT   1024*100/780
                            ;
        01F5    04  03  FA  GSCNUM: .BYTE   4,3, HOFA 0,0,0,0,0
   01F8  00     00  00
   01FB  00     00
                            ;
```

```
                                        .SBTTL  DSFONT--DISPLAY FONT CONVERTER
                                ;THIS ROUTINE SETS THE FONT TYPE WHEN ADDRESSED BY A PRINT STATEMENT
                                        .GLOBL  DSFONT
                                        .GLOBL  FIXNUM          ; FIXES FLOATING NUMBER FROM PRINT LIST
                                        .GLOBL  FONT            ; FONT GLOBAL FOR DISPLAY DRIVER
                                        .GLOBL  FPB             ; TEMPORARY STORAGE
                                ;
01FD    BD      000006          DSFONT: JSR     FIXNUM          ; GET NUMBER
0200    D7      006             STA B   FONT,D
0202    7F      000006          CLR     ERRCD           ; NO ERRORS!!!
0205    39                      RTS
```

```
                                        .SBTTL  DSFULL-PAGEFULL FUNCTION  2.26:
                                        .GLOBL  DSFULL
                                        .GLOBL  FULSTT
                                ;
                                ; 0 = NORMAL
                                ; 1 = HOME
                                ; 2 = PAGE
                                ; 3 = COPY & PAGE
                                ;
0206    BD      0000G           DSFULL: JSR     FIXNUM
0209    29      0A                      BVS     1$
0208    4D                              TST A
020C    26      07                      BNE     1$
020E    C1      03                      CMP B   3.1             ; OVERFLOW IGNORE IT
0210    22      03                      BHI     1$
0212    F7      0000G                   STA B   FULSTT
0215    7F      0000G           1$:     CLR     ERRCD
0218    39                              RTS
```

```
                                        .SBTTL  DSPLY--HOME AND PAG FUNCTIONS
                                ;THIS ROUTINES ACTUALLY SEND HOME AND PAGE TO ALL DEVICES ON THE IEC BUS
                                        .GLOBL  ADRDEV          ; ADDRESS DEVICE
                                        .GLOBL  SNDBFR          ; OUTPUT BUFFER
                                        .GLOBL  UNADR           ; UNADDRESS
                                        .GLOBL  DSPLY           ; DISPLAY FUNCTIONS
                                        .GLOBL  CRLF
                                        .GLOBL  BFRALC
                                ;
0219    86      00G     DSPLY:  LDA A   RTRNTG,I        ; TAG THAT
0218    36                      PSH A
021C    BD      0000G           JSR     ADRDEV          ; ADDRESS
021F    BD      0000G           JSR     BFRALC
0222    BD      0000G           JSR     CRLF
0225    BD      0000G           JSR     SNDBFR          ; SEND BUFFER
0228    BD      0000G           JSR     UNADR           ; UNADDRESS
0228    31                      INS                     ; REMOVE TAG
022C    39                      RTS                     ; RETURN
```

```
1                0001'                          .END
```

SYMBOL TABLE

```
ABRFLG=  0040        ADRDEV= XXXXXX G      AFAIL =  0030        AMAT  =  0010        ARRAY =  0020
ASTR  =  0020        ATLOD =  0008         ATSNTG= XXXXXX G     ATULD =  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     A9X   = XXXXXX G      BAKSTG= XXXXXX G     BANK  = XXXXXX G     BANKSW= XXXXXX G
BERRLG= XXXXXX G     BERSII=  0020         BLINK = XXXXXX G     BMAT  =  0004        BNKADR= XXXXXX G
BRKCNT= XXXXXX G     BSTR  =  0008         BUSACT=  0010        CDOPTR= XXXXXX G     CDSPTR= XXXXXX G
CHAR  = XXXXXX G     CHRCNT= XXXXXX G      CLPTR = XXXXXX G     CMAT  =  0001        COLCNT= XXXXXX G
CPYFLG= 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      CTLCHR   0000RG      CURSOR= XXXXXX G     DATDEV=  0022
DIMFLG=  0004        DIRCT =  0080         DISCNT= XXXXXX G     DISSRC=  0008        DL    = XXXXXX G
DOIT     0146R       DP    = XXXXXX G      DRBUSY= XXXXXX G     DREXTA= XXXXXX G     DREXTB= XXXXXX G
DSCOPY   0100R       DSDOIT   01CBR        DSDRAW   0180RG      DSFONT   01FDRG      DSFULL   0206RG
DSGDON   01E2R       DSGERR   01E0R        DSGEXT   01E1R       DSGRAF   0183RG      DSMOVE   0172RG
DSPCHR   0078RG      DSPCPY   0110RG       DSPCP2   0126RG      DSPDEV=  0020        DSPDON   00ECR
DSPEND   00E8R       DSPFHC   00E5R        DSPGO    00EER       DSPHOM   00E3R       DSPL     00F0R
DSPLY    0219RG      DSPOUT   00BCRG       DSPPGE   00DFR       DSPSTT= XXXXXX G     DSPWCT= XXXXXX G
DSRFR    0105R       DSRLCP   012CR        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     ERFBER= XXXXXX G      ERFILE= XXXXXX G     ERIOE = XXXXXX G     ERNLOG= XXXXXX G
ERNSEP= XXXXXX G     ERRCD = XXXXXX G      ERTERM= XXXXXX G     ERUNOF= XXXXXX G     ESTG  = XXXXXX G
EXTFLG=  0080        FILDEV=  0000         FIXNUM= XXXXXX G     FIX1  = XXXXXX G     FIX1A =  0003
FIX1B =  0004        FLASH    0060R        FMTWLD=  0008        FNELG =  0010        FONT  = XXXXXX G
FORTG = XXXXXX G     FPB   = XXXXXX G      FPMUL = XXXXXX G     FULDON   0070R       FULFNC   0035R
FULHOM   0033R       FULL  = XXXXXX G      FULPAG   002FR       FULSCN   0094RG      FULSTT= XXXXXX G
GENCUR   0141RG      GLBFLG= XXXXXX G      GOSTG = XXXXXX G     GSCNUM   01C5RG      HRDCPY=  0020
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      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        MTPD2 =  0023         NEWMOD   0157R       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
NULLTG= XXXXXX G     OBJA.R=  0002         OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000
OFFTIM=  0090        ONSFLG=  0002         ONTBL = XXXXXX G     ONTIM =  0015        OPRADR= XXXXXX G
OTHFUL   0038R       PAETG = XXXXXX G      PARM  =  0008        PCHAR = XXXXXX G     PCHARZ= XXXXXX G
PGCTR = XXXXXX G     PGMATR=  0002         PGMBP =  0005        PGMCD =  0009        PGMFP =  0003
PGMLEN=  0000        PGMLNN=  0007         PGMPTR= XXXXXX G     PGMTG = XXXXXX G     PIAHX = XXXXXX G
PIALX = XXXXXX G     PLOSTG= XXXXXX G      PNDEOF= XXXXXX G     PNDFLG= XXXXXX G     PNTIT    008FR
PNTNTG= XXXXXX G     PNTSTG= XXXXXX G      POINT = XXXXXX G     PPMODE= XXXXXX G     PRIDEF=  0001
PRTTG = XXXXXX G     PSCTG = XXXXXX G      PSHFPN= XXXXXX G     RECLFG=  0004        RFAIL =  00C0
RMAT  =  0040        RPTCTL=  0080         RSTR  =  0080        RTN      0076R       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        SCAN     0088R       SECDEF=  0002
SEMTTG= XXXXXX G     SETBNK= XXXXXX G      SNOBFR= XXXXXX G     SNOIT =  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     TMP1  = XXXXXX G      TMP2  = XXXXXX G     TRCFLG=  0020        TSIDEV=  0025
UNADR = XXXXXX G     UNDEF =  0080         VALTG = XXXXXX G     VALUND=  0040        VECTOR= XXXXXX G
WAIT  = XXXXXX G     WAIT1    0151R        WTHRU    0163R       WTHRU1   0173R       XAXIS = XXXXXX G
XFNBNK= XXXXXX G     YAXIS = XXXXXX G
```

```
DSPOUT DISPLAY OUTPUT CONTROL  RT-11 MMAC  VM02-10   14-OCT-76 01:30:48 PAGE 12+
SYMBOL TABLE

. ABS.    0000      00
          022D      01
ERRORS DETECTED: 0     WARNINGS POSTED: 0      FREE CORE: 2310. WORDS
,SY:DSPOUT/C<DK1:SEICLI,DSPOUT
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

SE1 1-3#

```
EEEEEEEEEE  VV      VV  LL          EEEEEEEEEE  NN      NN                  LL          SSSSSSSS   TTTTTTTTTT
EEEEEEEEEE  VV      VV  LL          EEEEEEEEEE  NNN     NN                  LL          SSSSSSSSSS TTTTTTTTTT
EE          VV      VV  LL          EE          NN N    NN                  LL          SS      S     TT
EE          VV      VV  LL          EE          NN NN   NN                  LL          SS            TT
EEEEEE      VV      VV  LL          EEEEEE      NN  NN  NN                  LL          SSSSSSSSS     TT
EEEEEE      VV      VV  LL          EEEEEE      NN  NN  NN                  LL           SSSSSSSSS    TT
EE           VV    VV   LL          EE          NN   NN NN                  LL                SS    TT
EE           VV    VV   LL          EE          NN    N NN                  LL          S     SS    TT
EEEEEEEEEE    VVVV    LLLLLLLLLL  EEEEEEEEEE  NN     NNN          .....   LLLLLLLLLL  SSSSSSSSSS   TT
EEEEEEEEEE     VV     LLLLLLLLLL  EEEEEEEEEE  NN      NN          ....    LLLLLLLLLL   SSSSSSSS    TT
```

TABLE OF CONTENTS

```
                                        .TITLE  EVLEN   EVALUATOR ENTRY POINTS
                                        .IDENT  /BE0029/
                                ;
                                        .GLOBL  EVLEN
0000'                           EVLEN   =.
                                        .GLOBL  PGMEVL,LNEVL,PUSHES,POPES,GETLNA
                                        .GLOBL  SRQRDY,SRQOFF,HLDEOF,DISKCL
                                        .GLOBL  PSHFPN,PULFPN,914X
                                        .GLOBL  SYSERR,SAFE,HALTR,SETERR
                                        .GLOBL  ANMCLD,EDTCLS
                                ;
                                        .SBTTL  *** FNEVL       USER FUNCTION CALL
                                        .GLOBL  FNEVL,QUSRFN
                                ;
                                ;       INPUTS
                                ;               TOKEN FOR FUNCTION IN CTKN
                                ;               ARGUMENT FOR FUNCTION ON STACK
                                ;
                                ;       OUTPUTS
                                ;               ALTERS FLOW OF CONTROL FOR FUNCTION CALL
                                ;
                                ;       NOTE.  THE OBJECT STATEMENT CAN'T HAVE GOSUB OR FOR SO STACK
                                ;          IS SAFE EVEN IF AN ERROR OCCURS.
                                ;
0000    96      00G             FNEVL:  LDA A   CTKN,D          ;CALC INDEX INTO FUNCTION TABLE
0002    80      00G                     SUB A   FNACOD,I
0004    48                              ASL A
0005    CE      0000G                   LDX     FNTBL,I         ;GET ADDR OF FNTBL FOR ADDR CALC
0008    BD      0000G                   JSR     LDXX            ;CALL DIRTY ADDR CALC AND LOAD
0008    27      1D                      BEQ     FNERR           ;IF ENTRY IS =0 FN IS NOT DEFINED
000D    DF      00G                     STX     R0,D            ;SAVE LINE PTR
000F    86      00G                     LDA A   RTRNTG,I        ;TAG RETURN ADDR
0011    36                              PSH A
0012    BD      0000G                   JSR     PUSHES          ;PUSH EVAL STATUS
0015    86      10                      LDA A   FNFLG,I         ;SET IN INTRPT PRCS MODE BIT
0017    97      00G                     STA A   LCLFLG,D
0019    DE      00G                     LDX     R0,D            ;FIX UP CLPTR NOW
001B    DF      00G                     STX     CLPTR,D
001D    BD      0000G                   JSR     914X            ;GET OF FIRST REAL CODE BYTE
0020    DF      00G                     STX     NTPTR,D
0022    BD      0000G                   JSR     LNEVL           ;CALL FOR EXE OF LINE GROUP
0025    BD      0000G           QUSRFN: JSR     POPES           ;PULL STATUS BACK OFF STACK
0028    32                              PUL A                   ;POP RETURN ADDR TAG
0029    39                              RTS                     ;RETURN TO CALLER
                                ;
002A    BD      0000G           FNERR:  JSR     SETERR
002D    00G                             .BYTE   ERNOFN
```

XXX DEF CALL TO DEFINE FN

```
                                        .SBTTL  XXX DEF         CALL TO DEFINE FN
                                        .GLOBL  DEF
                                ;
                                ;       INPUTS
                                ;               NEXT TOKEN IS USER FUNCTION TOKEN
                                ;               CLPTR IS POINTER TO CURRENT LINE
                                ;
                                ;       OUTPUTS
                                ;               FNTBL ENTRY FOR FUNCTION IS ALTERED TO POINT TO
                                ;                  THIS LINE.
                                ;
002E    96      00G     DEF:    LDA A   GLBFLG,D        ;NOT VALID IN CALC MODE
0030    85      C0              BIT A   192.,I
0032    27      22              BEQ     DEFERR
0034    DE      00G             LDX     NTPTR,D         ;CALC INDEX FOR THIS FUNCTION
0036    E6      03              LDA B   3,X
0038    C0      00G             SUB B   FNACOD.I
003A    58                      ASL B
003B    4F                      CLR A
003C    F8      005B'           ADD B   AFNTBL+1        ;ADD TABLE BASE ADDR
003F    89      005A'           ADC A   AFNTBL
0042    97      00G             STA A   R0,D            ;MOVE IT TO X REG
0044    D7      01G             STA B   R0+1,D
0046    DE      00G             LDX     R0,D
0048                            SEI
        0048    01      0F      .BYTE   01,17
004A    96      00G             LDA A   CLPTR,D         ;MOVE CURRENT LINE PTR TO FN CELL
004C    A7      00              STA A   0,X
004E    96      01G             LDA A   CLPTR+1,D
0050    A7      01              STA A   1,X
0052    0E                      CLI
0053    7E      0000G           JMP     HALTR           ;ABORT THIS LINE AND RTS
                                ;
0056    BD      0000G   DEFERR: JSR     SETERR
0059    00G                     .BYTE   ERNIMX          ;SET ERROR CODE AND RTS
                                ;
005A    0000G           AFNTBL: .WORD   FNTBL
```

```
                                        .SBTTL  *** TSTINT      TEST FOR INTERRUPTS PENDING
                                        .GLOBL  TSTINT
                                ;
                                ;       INPUTS
                                ;               FLAG BITS HAVE BEEN SET TO INDICATE PENDING INTERRUPTS
                                ;               PNDFLG FOR MOST INTERRUPTS
                                ;               PNDEOF DEFINES UNIT WITH EOF
                                ;               KEYSTK IS USED FOR USER FUNCTION KEYS
                                ;
                                ;       OUTPUTS
                                ;               CALLS TO ON UNIT ROUTINES OR ERROR EXITS
                                ;
005C  86  000G        TSTINT:   LDA A   RTRNTG,I        ;TAG RETURN ADDR
005E  36                        PSH A
005F  8D  0000G                 JSR     SAFE            ;SYSTEM CANE CALL ROM PACKS
0062                            SEI                     ;DISABLE INTERRUPTS
0062  01  0E            .BYTE   01,12
0064  96  000G                  LDA A   PNDFLG,D        ;GET FLAG BITS FOR PENDING INTERRUPTS
0066  26  50                    BNE     TSTON           ;ALL OTHER BITS ARE ON UNITS
0068  96  000G                  LDA A   PNDEOF,D        ;WAS EOF UNIT RAISED
006A  26  40                    BNE     TSTEOF          ;YES
006C  D6  000G                  LDA B   GLBFLG,D        ;KEYS DISABLED
006E  2A  04                    BPL     TSTSKP          ;IF IN CALC MODE DON'T LOOK AT KEYFLG
0070  C5  10                    BIT B   KEYFLG,I
0072  26  35                    BNE     TSTEXT
0074  F6  0000G       TSTSKP:  LDA B   KEYSTK          ;FUNCTION KEY SET
0077  27  30                    BEQ     TSTEXT          ;NO INTERRUPTS THIS TIME
0079  F7  0000G                 STA B   ANHOLD          ;ANY NONE ZERO WILL WORK
007C  CE  0001G                 LDX     KEYSTK+1,I      ;POP ONE ENTRY - IT IS IN ACC-A
007F  8D  0000G                 JSR     PSHFPN
0082  09                        DEX
0083  8D  0000G                 JSR     PULFPN
0086  0E                        CLI                     ;INTERRUPTS CAN OCCUR NOW
0087  58                        ASL B
0088  58                        ASL B
0089  D7  01G                   STA B   R0+1,D          ;SET UP TO CALL GETLNA
008B  7F  0000G                 CLR     R0
008E  8D  0000G                 JSR     GETLNA
0091  96  000G                  LDA A   ERRCD,D         ;IS ERROR CODE SET
0093  7F  0000G                 CLR     ERRCD           ;IF LINE NOT FOUND IGNORE KEY
0096  40                        TST A                   ;RESET COND CODE
0097  26  10                    BNE     TSTEXT          ;IF ERROR EXIT
0099  8D  0000G                 JSR     EDTCLS          ;GO CLOSE THE EDIT BUFFER
009C  8D  0000G                 JSR     PUSHES          ;STACK CURRENT STATUS
009F  DE  00G                   LDX     R0,D            ;SET UP NEXT LINE PTR
00A1  DF  00G                   STX     NLPTR,D
00A3  8D  0000G                 JSR     PGMEVL          ;EVAL KEY ROUTINE
00A6  8D  0000G                 JSR     POPES           ;RESTORE EVAL STATUS
00A9  0E                TSTEXT: CLI                     ;ALLOW INTERRUPTS AGAIN
00AA  32                        PUL A                   ;POP RETURN ADDRESS
00AB  39                        RTS
                                ;
00AC  B7  0000G         TSTEOF: STA A   HLDEOF          ;SAVE FOR ERROR ROUTINE
00AF  48                        ASL A                   ;DOUBLE UNIT TO GET INDEX
                                                        ; ALSO FORGET HIGH ORDER BIT
00B0  8B  10                    ADD A   16,I            ;BIAS IN TABLES
00B2  5F                        CLR B                   ;NO DISABLE BITS
```

```
0083  7F  0000G          CLR    PNDEOF       ;UNIT TAKEN
0086  20  24             BRA    ONRDY        ;JOIN COMMON CODE
                 ;
0088  4F         TSTON:  CLR A               ;SERVICE DISK
0089  BD  0000G          JSR    DISKCL
008C  96  00G            LDA A  PNDFLG,D
008E  27  E9             BEQ    TSTEXT       ;IF HE RESET IT STOP
00C0  CE  0159'          LDX    UNITBL,I     ;INDEX FOR DISABLE FLAGS
00C3  4F                 CLR A               ;INDEX FOR ON UNIT TABLE
00C4  C6  80             LDA B  128.,I       ;BIT TO TEST
00C6  D5  00G    ONLPA:  BIT B  PNDFLG,D     ;IS INTERRUPT BIT SET
00C8  26  06             BNE    ONHIT        ;TRA IF SO
00CA  08                 INX
00CB  4C                 INC A
00CC  4C                 INC A
00CD  54                 LSR B
00CE  20  F6             BRA    ONLPA
                 ;
00D0  D8  00G    ONHIT:  EOR B  PNDFLG,D     ;FLIP INTERRUPT PENDING FLAG
00D2  D7  00G            STA B  PNDFLG,D
00D4  E6  00             LDA B  0,X          ;GET INTERRUPT DISABLE FLAG
00D6  DA  00G            ORA B  GLBFLG,D
00D8  D7  00G            STA B  GLBFLG,D
00DA  E6  00             LDA B  0,X          ;NEED TO STACK FLAG ALSO
                 ;
00DC  0E         ONRDY:  CLI
00DD  36                 PSH A               ;PUSH ON UNIT INDEX
00DE  37                 PSH B
00DF  C6  00G            LDA B  ITM1TG,I
00E1  37                 PSH B
00E2  D6  00G            LDA B  GLBFLG,D     ;IF IN CALC MODE FORGET IT
00E4  2A  52             BPL    NOUNIT
00E6  CE  0000G          LDX    ONTBL,I      ;GET PTR TO ON UNIT LINE
00E9  BD  0000G          JSR    LDBX         ;LOAD HIGH BYTE INTO B
00EC  27  4A             BEQ    NOUNIT       ;LINE CAN'T BE IN PAGE ZERO
00EE  D7  00G            STA B  R0,D         ;SAVE HIGH HALF OF LINE ADDR
00F0  5F                 CLR B               ;RESET TABLE ENTRY (DISABLE INTERRUPT)
00F1  BD  0000G          JSR    STAX
00F4  4C                 INC A               ;GET NEXT BYTE
00F5  BD  0000G          JSR    LDBX
00F8  37                 PSH B               ;PUT HIGH BYTE ON STACK
00F9  D7  01G            STA B  R0+1,D       ;INTO R0 ALSO
00FB  5F                 CLR B
00FC  BD  0000G          JSR    STAX
00FF  96  00G            LDA A  R0,D         ;NOW STACK REST OF LINE ADDR
0101  36                 PSH A
0102  86  00G            LDA A  IMXTG,I      ;STACK TAG FOR LINE
0104  36                 PSH A
0105  BD  0000G          JSR    PUSHES       ;SAVE EVAL STATUS
0108  DE  00G            LDX    R0,D         ;GET LINE PTR
010A  DF  00G            STX    NLPTR,D
010C  BD  0000G          JSR    PGMEVL       ;EVAL ON UNIT LINE
010F  BD  0000G          JSR    POPES        ;RESTORE EVAL STATUS
0112  32                 PUL A               ;POP LINE TAG
0113  30                 TSX                 ;GET UNIT TABLE DISP FOR THIS ON UNIT
0114  A6  04             LDA A  4,X
0116  CE  0000G          LDX    ONTBL,I
```

```
        0119  33                      PUL A
        011A  BD   0000G              JSR    STAX        ;PUT HIGH BYTE INTO UNIT TABLE
        011D  4C                      INC A
        011E  33                      PUL B
        011F  BD   0000G              JSR    STAX
        0122  32              ONEXTA: PUL A              ,POP UNIT INDEX
        0123  33                      PUL B
        0124  32                      PUL A
        0125                          SEI                ;DISABLE INTERRUqTS
   0125  01   0F              .BYTE   01,17
        0127  53                      COM B              ;RESET INTERRUPT DISABLE BITS
        0128  D4   00G                AND B  GLBFLG,D
        012A  CA   02                 ORA B  OUAFLG,I    ;ON UNIT WAS RUN THIS CALL TO TSTINT
        012C  D7   00G                STA B  GLBFLG,D
        012E  81   06                 CMP A  6,I         ;SRQ
        0130  26   03                 BNE    ONEXTB
        0132  BD   0000G              JSR    SRQRDY
        0135  7E   00A9'      ONEXTB: JMP    TSTEXT
                              ;
        0138  81   04         NOUNIT: CMP A  4,I         ;IS IT PAGE FULL
        013A  27   E6                 BEQ    ONEXTP      ;IGNORE PAGE FULL
        013C  81   08                 CMP A  8.,I        ;EOI'S ARE NOT TOO BAD
        013E  27   E2                 BEQ    ONEXTA
        0140  BD   0000G              JSR    SRQOFF      ;KILL SRQ PENDING STATE
        0143  30                      TSX                ;GET ERROR CODE
        0144  A6   02                 LDA A  2,X
        0146  44                      LSR A
        0147  81   08                 CMP A  8.,I        ;TOP OUT AT EOF UNITS
        0149  23   02                 BLS    ONSKPZ
        014B  86   08                 LDA A  8.,I
        014D  8B   00G        ONSKPZ: ADD A  ERBRK,I
        014F  97   00G                STA A  ERRCD,D
        0151  96   00G                LDA A  ERRCDB,D    ;IF SET FORGET OLD ERRCD
        0153  27   CD                 BEQ    ONEXTA
        0155  97   00G                STA A  ERRCD,D     ;THIS IS REAL MESSAGE
        0157  20   C9                 BRA    ONEXTA
                              ;
        0159  00   00    00   UNITBL: .BYTE  0,0,0,DISSRQ,0,0,0,0
   015C  08   00   00
   015F  00   00
                              ;
              0001'                   .END
```

```
EVLEN  EVALUATOR ENTRY POINTS  RT-11 MMAC  VM02-10   14-OCT-76 01:31:03 PAGE 9+
SYMBOL TABLE

ABRFLG=  0040        AFAIL =  0030        AFNTBL   005AR       ALLOK =  0004        ALLTG = XXXXXX G
AMAT  =  0010        ANHOLD=  XXXXXX G    ARRAY =  0020       ASGCOD=  XXXXXX G    ASTR  =  0020
ATSNTG=  XXXXXX G    A14X  =  XXXXXX G    BAKSTG=  XXXXXX G   BMAT  =  0004        BSTR  =  0008
CALLTG=  XXXXXX G    CDOPTR=  XXXXXX G    CDSPTR=  XXXXXX G   CLPTR =  XXXXXX G    CMAT  =  0001
CONCOD=  XXXXXX G    CRCOD =  XXXXXX G    CSTR  =  0002       CTXN  =  XXXXXX G    DATCOD=  XXXXXX G
DEF      002ERG      DEFERR   0056R       DIMFLG=  0004        DISKCL=  XXXXXX G    DISSRQ=  0008
DREXTA=  XXXXXX G    DREXTB=  XXXXXX G    EDTCLS=  XXXXXX G   EJFCOD=  XXXXXX G    EOFTBL=  XXXXXX G
EOLTG =  XXXXXX G    EOSTG =  XXXXXX G    EQUCOD=  XXXXXX G   ERRSGN=  XXXXXX G    ERRRK =  XXXXXX G
ERDCMN=  XXXXXX G    EREOFN=  XXXXXX G    ERFORA=  XXXXXX G   ERLNNF=  XXXXXX G    ERNOT =  XXXXXX G
ERNIMX=  XXXXXX G    ERNOFN=  XXXXXX G    ERNXFN=  XXXXXX G   ERNXTA=  XXXXXX G    EROFA =  XXXXXX G
ERRCD =  XXXXXX G    ERRCOR=  XXXXXX G    ERSHAP=  XXXXXX G   ERSTOP=  XXXXXX G    ERUNDF=  XXXXXX G
ERVAL =  XXXXXX G    ERWSFL=  XXXXXX G    ESTG  =  XXXXXX G   EVLEN =  0000RG     EXTFLG=  0080
FNACOD=  XXXXXX G    FNERR    002AR       FNEVL    0000RG     FNFLG =  0010        FNTBL =  XXXXXX G
FORTG =  XXXXXX G    GETLNR=  XXXXXX G    GLBFLG=  XXXXXX G   GOSTG =  XXXXXX G    HOLTR =  XXXXXX G
HLDEOF=  XXXXXX G    IMACOD=  XXXXXX G    IMXFLG=  0020        IMXTG =  XXXXXX G    ITM1TG=  XXXXXX G
ITM2TG=  XXXXXX G    JMPAX =  XXXXXX G    JMPX  =  XXXXXX G   KEYFLG=  0010        KEYSTK=  XXXXXX G
LBRKTG=  XXXXXX G    LCLFLG=  XXXXXX G    LDAX  =  XXXXXX G   LDBX  =  XXXXXX G    LDXX  =  XXXXXX G
LISTTG=  XXXXXX G    LITCOD=  XXXXXX G    LNEVL =  XXXXXX G   LNNOTG=  XXXXXX G    LSP   =  XXXXXX G
LSTCOD=  XXXXXX G    MMICOD=  XXXXXX G    MPLCOD=  XXXXXX G   MULCOD=  XXXXXX G    NCSFLG=  0001
NLPTR =  XXXXXX G    NOUNIT   0138R       NTAPTR=  0008        NTATTR=  0004        NTDIMS=  0009
NTDLEN=  0005        NTLINK=  0000        NTNAME=  0002        NTPTR =  XXXXXX G    NTSPTR=  0008
NTVAL =  0005        NTWCOL=  0007        NTWLEN=  0007        NTWROW=  0005        NULLTG=  XXXXXX G
OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        ONEXTA   0122R
ONEXTB   0135R       ONHIT    000OR       ONLPA    00C6R       ONRDY    000CR       ONSFLG=  0002
ONSKPZ   014DR       ONTBL =  XXXXXX G    OPRADR=  XXXXXX G    OUAFLG=  0002        PAETG =  XXXXXX G
PARM  =  0008        PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMEVL=  XXXXXX G
PGMFP =  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
POPES =  XXXXXX G    PRITG =  XXXXXX G    PSCTG =  XXXXXX G   PSHFPN=  XXXXXX G    PULFPN=  XXXXXX G
PUSHES=  XXXXXX G    QUSRFN   0025RG      RFAIL =  00C0        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    SAFE  =  XXXXXX G
SBP   =  XXXXXX G    SCALER=  0040        SEMITG=  XXXXXX G   SETERR=  XXXXXX G    SIZCOD=  XXXXXX G
SRQOFF=  XXXXXX G    SRQRDY=  XXXXXX G    STAX  =  XXXXXX G   STPFLG=  0040        STPIR =  XXXXXX G
STRING=  0010        SYSERR=  XXXXXX G    TRCFLG=  0020        TSTEOF   00ACR       TSTEXT   00A9R
TSTINT   005CRG      TSTON    0088R       TSTSKP   0074R       UNDEF =  0080        UNITBL   0159R
VALERR=  0040        VALTG =  XXXXXX G

. ABS.   0000    00
         01a1    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2619  WORDS
.SY: EVLEN/C<DK1: SETCL1.EVLEN
```

CROSS REFERENCE TABLE (CREF V01-03)

	2-5				
A14X	7-8#	7-39			
ABRFLG	2-10#				
AEQIL	5-38#				
AFNTBL	8-20	8-21	8-36#		
ALLOK	4-13#				
ALLTG	5-24#				
AMAT	5-37#				
ANMOLD	7-10#	9-27*			
ARRAY	4-10#				
ASGCOD	1-36#				
ASTR	5-36#				
ATSNTG	5-22#				
BAKSTG	5-23#				
BMAT	5-40#				
BSTR	5-39#				
CALLTG	5-27#				
CDOPTR	3-40#				
CDSPTR	3-39#				
CLPTR	2-4#	7-38*	8-26	8-28	
CMAT	5-42#				
CONCOD	1-45#				
CRCOD	1-44#				
CSTR	5-41#				
CTKN	2-7#	7-25			
DATCOD	1-47#				
DEF	8-2#	8-12#			
DEFERR	8-14	8-33#			
DIMFLG	2-13#				
DISKCL	7-7#	9-61			
DISSRQ	2-21#	9-150			
DREXTA	2-25#				
DREXTB	2-26#				
EDTCLS	7-10#	9-42			
EOFCOD	1-41#				
EOFTBL	3-5#				
EOLTG	5-28#				
EOSTG	5-29#				
EQUCOD	1-37#				
ERASGN	6-16#				
ERBRK	6-14#	9-143			
ERDOMN	6-7#				
EREOFN	6-15#				
ERFORA	6-11#				
ERLNNF	6-8#				
ERNOT	6-13#				
ERNTMX	6-5#	8-34			
ERNOFN	6-4#	7-47			
ERNXFN	6-20#				
ERNXTA	6-12#				
EROFA	6-10#				
ERRCD	2-31#	9-38	9-39*	9-144*	9-147*
ERRCDB	2-32#	9-145			
ERSHAP	6-18#				
ERSTOP	6-9#				
ERUNOF	6-17#				

CROSS REFERENCE TABLE (CREF V01-03.)

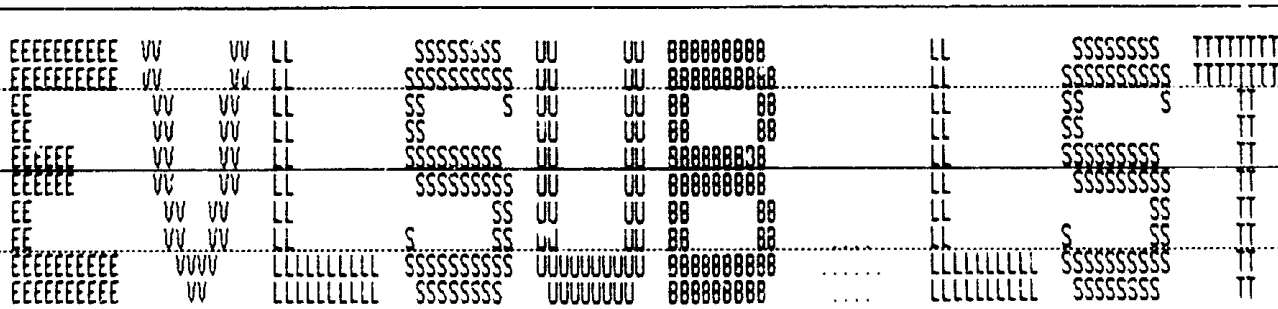
CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SIZCOD	1-79#					
SRQOFT	7-7#	9-136				
SRQRDY	7-7#	9-129				
STBX	3-30#	9-94	9-100	9-115	9-118	
STPFLG	2-18#					
STPTR	2-51#					
STRING	4-11#					
SYSERR	7-9#					
TRCFLG	2-19#					
TSTEOF	9-20	9-52#				
TSTEXT	9-24	9-26	9-41	9-48#	9-63	9-130
TSTINT	9-2#	9-13#				
TSTON	9-18	9-60#				
TSTSKP	9-22	9-25#				
UNDEF	4-8#					
UNLTBL	9-64	9-150#				
VALERR	4-29#					
VALTG	5-16#					

CROSS REFERENCE TABLE (CREF V01-02)

PLRA 1-20#
PLRB 1-27#
SE1 1-3# 8-25 9-16 9-122



14-OCT-76

TABLE OF CONTENTS

```
                                        .TITLE  EVLSUB  EVALUATOR SUBROUTINES
                                        .IDENT  /BE5024/
                                ;
                                        .GLOBL  EVLSUB
00C0'                           EVLSUB  =       .
                                        .GLOBL  FIX1,SYSERR,DISBLE,ENABLE
                                        .GLOBL  A5X,A8X,A9X,PULFPN,PSHFPN,ZX
                                        .GLOBL  PSHRET,RTRN
                                ;
                                        .SBTTL  ***  TYPARG      TYPE OF ARGUMENT
                                        .GLOBL  TYPARG,SETARG,CLRARG
                                ;
                                ;       TYPARG
                                ;
                                ;       INPUTS
                                ;               R0 IS CURRENT STACK LOCATION-1 OF NEXT TAG TO USE
                                ;
                                ;       OUTPUTS
                                ;               R0 IS UPDATED TO POINT TO NEXT TAG-1.
                                ;               R4 HAS FLAG BITS SET
                                ;               ACC-A ALSO HAS A COPY OF R4 FLAG BITS
                                ;
                                ;       RR AA  BB CC    IS FORMAT OF OUTPUT BITS
                                ;
                                ;       RR - RESULT AREA CLASS
                                ;       AA - OPERAND CLASS OF LAST OPERAND TESTED
                                ;       BB - SECOND OPERAND
                                ;       CC - THIRD OPERAND
                                ;
                                ;       00 = SCALER
                                ;       01 = ARRAY
                                ;       10 = STRING
                                ;       11 - ERROR, INPUT IS UNDEFINED, OUTPUT CAN'T RECIEVE DATA
                                ;
0000    30                      SETARG: TSX                     ;SET UP FOR THIS ENTRY POINT TO TYPARG
0001    08                              INX
0002    DF      00G                     STX     R0,0
0004    7F      0000G           CLRARG: CLR     R4
0007    74      0000G           TYPARG: LSR     R4              ;AGE OPERANDS IN CLASS
000A    74      0000G                   LSR     R4
000D    DE      00G                     LDX     R0,0            ;GET NEXT TAG
000F    A6      01                      LDA A   1,X
0011    81      00G                     CMP A   VALTG,I         ;VALUE ON STACK
0013    27      17                      BEQ     TAVAL
0015    81      00G                     CMP A   PNTNTG,I        ;N. T. NUMERIC
0017    27      1D                      BEQ     TANTN
0019    81      00G                     CMP A   PAETG,I         ;ARRAY ELEMENT
001B    27      46                      BEQ     TAPAE
001D    81      00G                     CMP A   PNTSTG,I        ;N. T. STRING
001F    27      20                      BEQ     TANTS
0021    81      00G                     CMP A   PLOSTG,I        ;LITERAL IN OBJECT STRING
0023    27      04                      BEQ     TAJMP
0025    31      00G                     CMP A   PSCTG,I         ;HE MEY BE RESCANNING THE STACK
0027    26      61                      BNE     TAUOPS
0029    7E      00C3'           TAJMP:  JMP     TALOS
                                ;
002C    DE      00G             TAVAL:  LDX     R0,0            ;UPDATE R0 TO NEXT TAG
```

XXX TYPARG TYPE OF ARGUMENT

```
                                JSR     A9X
                                STX     R0,D
                                LDA A   R4,D            ;RETURN R4 BITS
                                RTS                     ;RETURN TO CALLER
                        ;
TANTN:                  LDX     2,X             ;GET N. T. PTR
                        LDA A   NTATTR,X        ;IS IT MATRIX
                        BIT A   ARRAY,I
                        BNE     TAMAT
                        BIT A   UNDEF,I         ;IF DEFINED IT IS A SCALER
                        BNE     TAUNDF
                        STX     R1,D            ;NEED TO MOVE SCALER
                        LDX     R0,D
                        STX     R2,D
                        LDA A   8.,I            ;BYTE COUNT TO MOVE
TALPA:                  LDX     R1,D            ;INPUT DATA IN N. T.
                        LDA B   NTVAL,X
                        INX
                        STX     R1,D
                        LDX     R2,D            ;OUTPUT INTO STACK
                        STA B   2,X
                        INX
                        STX     R2,D
                        DEC A                   ;LOOP FOR 8 BYTES
                        BNE     TALPA
                        ;
TATGVL:                 LDA A   VALTG,I         ;TAG AS VALUE ON STACK NOW
                        LDX     R0,D
                        STA A   1,X
                        BRA     TAVAL           ;UPDATE R0 AND EXIT
                        ;
TAPAE:                  STX     R2,D            ;STACK IS OUTPUT AREA
                        LDX     4,X             ;GET POINTER TO DATA
                        STX     R1,D
                        LDA A   0,X             ;TEST FOR UNDEFINED VALUE
                        BIT A   VALERR,I
                        BNE     TAUNDF
                        LDA A   8.,I
TALP8:                  LDX     R1,D            ;GET INPUT BYTE
                        LDA B   0,X
                        INX
                        STX     R1,D
                        LDX     R2,D            ;PUT IT IN OUTPUT AREA
                        STA B   2,X
                        INX
                        STX     R2,D
                        DEC A                   ;LOOP CONTROL
                        BNE     TALP8
                        BRA     TATGVL          ;TAG DATA AND FIX UP R0
                        ;
TAMAT:                  BRA     TAMAT2          ;KEEP LOOKING
                        ;
TAUNDF:                 LDA A   ERUNDF,I        ;UNDEFINED DATA FOUND
                        STA A   ERRCD,D
                        ;
TAOOPS:                 LDA A   AFAIL,I         ;SET ERROR BITS
                        ORA A   R4,D
```

```
008E  97  00G                STA A  R4,D         ;NOTE - R4 BITS ARE IN ACC-A
0090  39                     RTS                 ;GOOD LUCK NEXT TIME
                    ;
0091  EE  02        TANTS:   LDX    2,X          ;GET PTR TO N.T. ENTRY
0093  86  80                 LDA A  UNDEF,I      ;IS STRING DEFINED
0095  A5  04                 BIT A  NTATTR,X
0097  26  ED                 BNE    TAUNDF
0099  EE  08                 LDX    NTSPTR,X     ;CAN BE UNDEF STILL
009B  A6  02                 LDA A  OBJATR,X     ;SEE IF IT IS DEFINED
009D  85  40                 BIT A  VALERR,I
009F  26  E5                 BNE    TAUNDF
00A1  DE  00G                LDX    R0,D         ;RESTORE X REG.
00A3  EE  02                 LDX    2,X
00A5  A6  08                 LDA A  NTSPTR,X     ;GET AND FIX UP STRING ADDR
00A7  E6  0C                 LDA B  NTSPTR+1,X
00A9  CB  03                 ADD B  OBJDT-2,I
00AB  89  00                 ADC A  0,I
00AD  EE  07                 LDX    NTWLEN,X     ;MOVE WORKING LENGTH TO R1 FOR SC ENTRY
00AF  DF  00G                STX    R1,D
00B1  DE  00G                LDX    R0,D         ;GET STACK ENTRY ADDR
00B3  A7  04                 STA A  4,X          ;FIX UP DATA ADDR
00B5  E7  05                 STA B  5,X
00B7  EE  04                 LDX    4,X          ;NOW PUT WORKING LENGTH INTO DATA OBJECT
00B9  96  00G                LDA A  R1,D
00BB  A7  00                 STA A  0,X
00BD  96  01G                LDA A  R1+1,D
00BF  A7  01                 STA A  1,X
00C1  DE  00G                LDX    R0,D         ;RESTORE STACK POINTER
                    ;
00C3  86  00G       TALOS:   LDA A  PSCTG,I      ;TAG AS STRING COUNT
00C5  A7  01                 STA A  1,X
00C7  DE  00G                LDX    R0,D         ;UPDATE PSUEDO STACK POINTER
00C9  BD  0000G              JSR    A5X
00CC  DF  00G                STX    R0,D
00CE  86  20                 LDA A  ASTR,I       ;MARK R4 FOR STRING ENTRY
00D0  9A  00G                ORA A  R4,D
00D2  97  00G                STA A  R4,D
00D4  39                     RTS
                    ;
00D5  86  10        TAMAT2:  LDA A  AMAT,I       ;MARK IT AS ARRAY
00D7  9A  00G                ORA A  R4,D
00D9  97  00G                STA A  R4,D
00DB  DF  00G                STX    R1,D         ;SAVE NT PTR
00DD  BD  018A'              JSR    MATSIZ       ;CALC END ADDROF DATA
00E0  DF  00G                STX    R2,D         ;SAVE END ADDR
00E2  DE  00G                LDX    R1,D         ;GO BACK TO NT ENTRY
00E4  A6  04                 LDA A  NTATTR,X     ;GET ATTR BYTE TO SEE IF ALL DATA DEFINED
00E6  85  04                 BIT A  ALLOK,I
00E8  27  03                 BEQ    TAMSKB       ;I MUST TEST IT NOW
00EA  7E  002C'     TAMEND:  JMP    TAVAL        ;GO WAY BACK
00ED  EE  08        TAMSKB:  LDX    NTAPTR,X     ;GET DATA BASE ADDR
00EF  C6  40                 LDA B  VALERR,I
00F1  E5  05        TAMLP:   BIT B  5,X          ;GET FIRST EXP BYTE (SAVE ACC-A)
00F3  26  0F                 BNE    TAMFL        ;TOO BAD
00F5  BD  0000G              JSR    A8X          ;TRY NEXT ONE
00F8  9C  00G                CPX    R2,D
00FA  26  F5                 BNE    TAMLP
```

```
00FC   DE   00G              LDX     R1,D          ;GET NT PTR
00FE   89   04               ORA A   ALLOK,I       ;IT IS ALL THERE NOW
0100   A7   04               STA A   NTATTR,X
0102   20   E6               BRA     TAMEND
                     ;
0104   7E   0086'    TAMFL:  JMP     TAUNDF        ;I FOUND UNDEFINED DATA
```

```
                                        .SBTTL  *** TYPRES      TYPE OF RESULT AREA
                                        .GLOBL  TYPRES
                                ;
                                ;       INPUTS
                                ;               R0 HAS ADDR OF CURRENT TAG-1
                                ;
                                ;       OUTPUT
                                ;               BITS SET IN R4 FOR ARRAY/STRING OUTPUT AREA
                                ;               ACC-A IS LOADED WITH R4 BITS ALSO
                                ;
0107  86  000         TYPRES: LDA A   RTRNTG,I        ;TAG IN CASE OF CALL TO STR DIM
0109  36                      PSH A
010A  DE  000                 LDX     R0,D            ;GET CURRENT TAG ADDR
010C  A6  01                  LDA A   1,X             ;GET TAG
010E  81  000                 CMP A   PAETG,I         ;AN ARRAY ELEM CAN RECEIVE DATA
0110  27  25                  BEQ     TROKA
0112  81  000                 CMP A   PNINTG,I
0114  27  0C                  BEQ     TRPNTN
0116  81  000                 CMP A   PNTSTG,I        ;IS IT A STRING
0118  27  2C                  BEQ     TRSTR
                                ;
011A  32              TROOPS: PUL A                   ;PULL RETURN TAG
011B  86  C0                  LDA A   #FAIL,I         ;SET ERROR FLAG
011D  9A  000                 ORA A   R4,D
011F  97  000                 STA A   R4,D            ;NOTE - ACC-A HAS R4 BITS
0121  39                      RTS                     ;CALLER WILL SET ERROR CODE
                                ;
0122  EE  02          TRPNTN: LDX     2,X             ;GET ADDR N.T. ENTRY
0124  A6  04                  LDA A   NTATTR,X
0126  85  40                  BIT A   SCALER,I        ;IS IT SCALER
0128  26  0D                  BNE     TROKA
012A  85  20                  BIT A   ARRAY,I         ;IS MATRIX DEFINED
012C  27  EC                  BEQ     TROOPS
012E  86  40                  LDA A   RMAT,I          ;MARK R4 FOR MATRIX
0130  9A  000                 ORA A   R4,D
0132  97  000                 STA A   R4,D
0134  BD  018A'              JSR     MATSIZ
                                ;
0137  DE  000         TROKA:  LDX     NTPTR,D         ;NEXT TOKEN MUST BE ASSIGN
0139  86  000                 LDA A   ASGCOD,I
013B  A1  00                  CMP A   0,X
013D  26  08                  BNE     TROOPS          ;TRA IF IT WAS NOT ASSIGN
013F  08                      INX                     ;SKIP ASSIGN TOKEN
0140  DF  000                 STX     NTPTR,D
0142  32                      PUL A                   ;PULL RETURN TAG
0143  96  000                 LDA A   R4,D            ;RETURN R4 BITS IN ACC-A
0145  39                      RTS
                                ;
0146  86  80          TRSTR:  LDA A   #STR,I          ;SET R4 FLAGS FOR STRING
0148  9A  000                 ORA A   R4,D
014A  97  000                 STA A   R4,D
014C  EE  02                  LDX     2,X             ;GET PTR TO N.T.
014E  A6  04                  LDA A   NTATTR,X        ;IS IT DEFINED STRING
0150  85  80                  BIT A   #UNDEF,I
0152  27  E3                  BEQ     TROKA           ;IT IS OKAY
0154  BD  0000G               JSR     DISBLE          ;SAFE FROM INTERRUPTS
0157  DE  000                 LDX     R0,D            ;GET N.T. PTR AGAIN
```

```
   58  0159  EE  02      LDX     2,X
   59  015B  86  90      LDA A   UNDEF+STRING.,I  ;SET UP THE STRING IN STACK SPACE
   60  015D  A7  04      STA A   NTATTR,X
   61  015F  6F  05      CLR     NTDLEN,X         ;LENGTH IS 72
   62  0161  86  48      LDA A   72.,I
   63  0163  A7  06      STA A   NTDLEN+1,X
   64  0165  6F  07      CLR     NTWLEN,X         ;LENGTH TO 0 IN CASE OF ERROR LATER
   65  0167  6F  08      CLR     NTWLEN+1,X
   66  0169  96  00G     LDA A   LSP,D            ;GET LOW STACK POINTER
   67  016B  D6  01G     LDA B   LSP+1,D
   68  016D  A7  08      STA A   NTSPTR,X         ;PUT STRING ADDR INTO N. T.
   69  016F  E7  0C      STA B   NTSPTR+1,X
   70  0171  CB  4D      ADD B   77.,I            ;UPDATE LSP
   71  0173  89  00      ADC A   0,I
   72  0175  97  00G     STA A   LSP,D
   73  0177  D7  01G     STA B   LSP+1,D
   74  0179  EE  08      LDX     NTSPTR,X         ;GET POINTER TO MEMORY
   75  017B  6F  00      CLR     0,X              ;SET UP COUNT
   76  017D  86  4D      LDA A   77.,I
   77  017F  A7  01      STA A   1,X
   78  0181  86  60      LDA A   96.,I            ;ATTR BYTE UNDEF STRING
   79  0183  A7  02      STA A   2,X
   80  0185  BD  0000G   JSR     ENABLE           ;INTERRUPTS ARE OKAY NOW
   81  0188  20  AD      BRA     TROKA            ;I HAVE A REAL STRING NOW
```

```
                                        .SBTTL  *** MATSIZ      MATRIX SIZE CALCULATIONS
                                        .GLOBL  MATSIZ
                                ;
                                        .GLOBL  TABPNT,T1,T2,T3
                                ;
                                ;       CALCULATE SPACE FOR A MATRIX AND RETURN ORIGIN AND END ADDRESSES.
                                ;
                                ;       INPUTS
                                ;       X - ADDR OF NAME TABLE ENTRY
                                ;       R0 - ADDR OF STACK ENTRY -1
                                ;
                                ;       OUTPUTS
                                ;       X - ADDR OF END OF DATA SPACE-5
                                ;       STACK ENTRY GETS ROW COUNT,COL COUNT & END OF DATA PTR
                                ;
                                ;       WORKING REGISTERS
                                ;       TABPNT - TEMP SAVE AREA FOR NAME TABLE POINTER
                                ;       T1 & T2 - TEMP REG FOR 16 BIT OPERAND
                                ;       T3 - TEMP REG FOR 8 BIT OPERAND
                                ;
                                ;       NOTE.... ONE OF THE OPERANDS MUST BE ONLY 8 BITS LONG.
                                ;
018A  DF  00G   MATSIZ: STX     TABPNT,D        ;SAVE NT PTR
018C  A6  05            LDA A   NTWRCW,X
018E  E6  06            LDA B   NTWROW+1,X
0190  DE  00G           LDX     R0,D            ;NOW PUT IT IN STACK ENTRY
0192  A7  06            STA A   6,X
0194  E7  07            STA B   7,X
0196  DE  00G           LDX     TABPNT,D        ;BACK TO NAME TABLE
0198  A6  07            LDA A   NTWCOL,X        ;GET COLUMN SIZE
019A  E6  08            LDA B   NTWCOL+1,X
019C  DE  00G           LDX     R0,D            ;PUT COL ON STACK
019E  A7  08            STA A   8.,X
01A0  E7  09            STA B   9.,X
01A2  EE  08            LDX     8.,X            ;IF ROWS = 0 SET TO 1
01A4  26  04            BNE     MSSKP
01A6  DE  00G           LDX     R0,D            ;RESTORE STACK PTR
01A8  CC  09            INC     9.,X
01AA  DE  00G   MSSKP:  LDX     TABPNT,D        ;GET NT PTR
01AC  EE  05            LDX     NTWROW,X        ;ASSUME THIS IS 16 BIT OPERAND
01AE  DF  00G           STX     T1,D            ;T1 & T2 ARE BIG OPERAND
01B0  DE  00G           LDX     TABPNT,D        ;BACK TO N.T.
01B2  A6  07            LDA A   NTWCOL,X
01B4  26  34            BNE     MSFLIP          ;I HAVE OPERANDS BACKWARDS IF NOT EQ 0
01B6  E6  08            LDA B   NTWCOL+1,X
01B8  26  08            BNE     MSCONT          ;VECTOR SIZE IS ROW#8
01BA  96  00G           LDA A   T1,D
01BC  D6  00G           LDA B   T2,D
01BE  58                ASL B
01BF  49                ROL A
01C0  20  15            BRA     MSBY4
                ;
01C2  D7  00G   MSCONT: STA B   T3,D            ;T3 IS 8 BIT OPERAND
01C4  4F                CLR A                   ;PRESET ACCUM TO ZERO
01C5  5F                CLR B
01C6  0D                SEC                     ;SO I WILL LOOP 8 TIMES
```

```
    58  01C7  79  0000G           ROL   T3              ;SHIFT AND TEST BIT
    59  01CA  24  04        MSLP: BCC   MSNOADD         ;NO BIT IS NO ADD
    60  01CC  DB  00G             ADD B T2,D
    61  01CE  99  00G             ADC A T1,D
    62  01D0  58                  ASL B                 ;SHIFT RESULT FOR MULT BY 2
    63  01D1  49                  ROL A
    64  01D2  78  0000G           ASL   T3              ;TEST NEXT MULTIPLIER BIT
    65  01D5  26  F3              BNE   MSLP            ;LOOP UNTIL I USE ALL BITS
    66  01D7  58              MSBY4: ASL B
    67  01D8  49                  ROL A
    68  01D9  58                  ASL B
    69  01DA  49                  ROL A
    70  01DB  DE  00G             LDX   TABPNT,D        ;GET PTR TO NAME TABLE
    71  01DD  EB  0C              ADD B NTAPTR+1,X      ;ADD MEMORY BASE ADDRESS
    72  01DF  A9  08              ADC A NTAPTR,X
    73  01E1  DE  00G             LDX   R0,D            ;GET STACK POINTER
    74  01E3  A7  04              STA A 4,X
    75  01E5  E7  05              STA B 5,X
    76  01E7  EE  04              LDX   4,X             ;RETURN END ADDR IN X
    77  01E9  39                  RTS
    78                        ;
    79  01EA  EE  07        MSFLIP: LDX   NTWCOL,X
    80  01EC  D6  00G             LDA B T2,D            ;GET 8 BIT OPERAND
    81  01EE  DF  00G             STX   T1,D
    82  01F0  20  00              BRA   MSCONT
```

```
                                        .SBTTL  XXX PUSHES      PUSH EVALUATOR STATUS
                                        .GLOBL  PUSHES
                                        ;
                                        ;       PUSH EVAL CONTROLS ON STACK IN PREPERATION FOR A RECURSION.
                                        ;       ALSO SET CONTROLS TO ZERO.
                                        ;
01F2                                    PUSHES: PLRB                    ;SAVE RETURN ADDR
01F3    CE    0007G                             LDX     CLPTR+7,I       ;LAST BYTE FIRST
01F6    A6    00                        PUSHLP: LDA A   0,X
01F8    36                                      PSH A
01F9    6F    00                                CLR     0,X
0200    09                                      DEX
0201    8C    FFFFG                             CPX     CLPTR-1,I
0204    26    F5                                BNE     PUSHLP
0206    86    00G                               LDA A   ESTG,I
0208    36                                      PSH A
0209    7E    0000G                             JMP     DREXTB          ;RETURN TO CALLER
```

```
                                        .SBTTL  XXX POPES       POP EVALUATOR STATUS
                                        .GLOBL  POPES
                                ;
                                ;       POP EVAL CONTROLS FROM STACK FOR RETURN FROM RECURSION.
                                ;
020C    CE      0000G   POPES: LDX     CLPTR,I         ;ENTRY IS 8 BYTS - OR FPN SIZE
020F    7E      0000G           JMP     PULFPN          ;DIRTY BUT SHORT
```

```
                                        .SBTTL  *** GETLN/GETLNA       GET A LINE PTR GIVEN A LINE NUMBER
                                        .GLOBL  GETLN,GETLNA
                                ;
                                ;       INPUTS
                                ;               VALUE ON STACK IS LINE NO.  (GETLN)
                                ;               VALUE IN R0 IS LINE NO.  (GETLNA)
                                ;
                                ;       OUTPUTS
                                ;               R0 HAS POINTER TO LINE
                                ;
                                ;       IF ERRORS- LINE NOT FOUND
                                ;               R0 HAS ZERO
                                ;               ERRCD HAS CODE FOR LINE NOT FOUND ERROR
                                ;
0212  BD  0000G   GETLNA: JSR     PSHRET
0215  20  12              BRA     GETLNX          ;JOIN COMMON CODE
                                ;
0217  BD  0000G   GETLN:  JSR     PSHRET
021A  30                  TSX                     ;SET UP FOR FIX1
021B  BD  0000G           JSR     FIX1
021E  26  4E              BNE     GLFAIL
0220  EE  03              LDX     3,X             ;GET LINE NO.
0222  DF  00G             STX     R0,D
0224  30                  TSX                     ;PRUNE STACK
0225  BD  0000G           JSR     A9X
0228  35                  TXS
0229  96  00G   GETLNX: LDA A   GLBFLG,D   ;IF IN CALC MODE USE FOR LIST
022B  2A  04              BPL     GETIMX
022D  DE  00G             LDX     CLPTR,D         ;START WHERE WE ARE
022F  26  04              BNE     GLSKIP          ;IF NO LINE START AT FIRST
0231  DE  00G   GETIMX: LDX     PGMPTR,D
0233  27  35              BEQ     GLFAIL          ;NO PROGRAM - FORGET IT
0235  96  00G   GLSKIP: LDA A   R0,D             ;GET GIVEN LINE NUMBER
0237  D6  01G             LDA B   R0+1,D
0239  A1  07              CMP A   PGMLNN,X        ;GIVEN LINE NO. TO CURRENT ONE
023B  22  0A              BHI     GLFLP           ;IF LINE NO. IS HIGH LOOK FOWARD
023D  25  1A              BCS     GLBLP           ;IF LOW LOOK BACK INTO PROGRAM
023F  E1  08              CMP B   PGMLNN+1,X
0241  22  04              BHI     GLFLP
0243  25  14              BCS     GLBLP
0245  20  22              BRA     GLHIT           ;IF EQ I HAVE RIGHT LINE
                                ;
0247  EE  03      GLFLP:  LDX     PGMFP,X         ;CHAIN FOWARD
0249  27  23              BEQ     GLFAIL          ;IF END LINE SEARCH FAILED
024B  A1  07              CMP A   PGMLNN,X
024D  22  F8              BHI     GLFLP
024F  25  1D              BCS     GLFAIL
0251  E1  08              CMP B   PGMLNN+1,X
0253  22  F2              BHI     GLFLP
0255  25  17              BCS     GLFAIL
0257  20  10              BRA     GLHIT
                                ;
0259  EE  05      GLBLP:  LDX     PGMBP,X         ;BACK UP ONE LINE
025B  27  11              BEQ     GLFAIL          ;IF END ERROR
025D  A1  07              CMP A   PGMLNN,X
025F  22  0D              BHI     GLFAIL
0261  25  F6              BCS     GLBLP
```

*** GETLN/GETLNA GET A LINE PTR GIVEN A LINE NUMBER

```
58   0263  E1  08          CMP B  PGMLNN+1,X
59   0265  22  07          BHI    GLFAIL
60   0267  25  F0          BCS    GLBLP
61                         BRA    GLHIT          ;FALL INTO HIT ROUTINE
62                         ;
63   0269  DF  00G   GLHIT: STX    R0,D           ;RETURN LINE ADDRESS
64   026B  7E  0000G        JMP    RTRN           ;RETURN TO CALLER
65                         ;
66   026E  86  00G   GLFAIL: LDA A  ERLNF 1        ;ERROR - LINE NOT FOUND
67   0270  97  00G          STA A  ERRCD,X         ;SET ERROR CODE
68   0272  DE  00G          LDX    ZX,D            ;CLEAR PTR
69   0274  DF  00G          STX    R0,D
70   0276  7E  0000G        JMP    RTRN           ;BETTER LUCK NEXT TIME
```

```
                                        .SBTTL  XXX LOCTG       LOCATE A GIVEN TAG STARTING AT R0
                                        .GLOBL  LOCTG
                                ;
                                ;       INPUTS
                                ;               TAG IS IN ACC-A
                                ;               STARTING STACK ADDR-1 IS IN R0
                                ;
                                ;       OUTPUTS
                                ;               R0 IS ZERO IF TAG NOT FOUND
                                ;               R0 IS STACK ADDR-1 IF FOUND
                                ;
0279  97    00G     LOCTG:  STA A   R1,D            ;SAVE TAG TO BE FOUND
027B  DE    00G     LTLP:   LDX     R0,D            ;GET STARTING LOCATION-1
027D  A6    01              LDA A   1,X             ;GET TAG FROM STACK
027F  91    00G             CMP A   R1,D            ;IS IT THE ONE HE NEEDS
0281  27    0E              BEQ     LCTGEX          ;I FOUND IT
0283  81    00G             CMP A   EOSTG,I         ;IS IT END OF STACK
0285  27    05              BEQ     LCTGZ           ;IF EOS ZERO R0 & RETURN
0287  BD    02AD'           JSR     BACKUP          ;GO BACK IN STACK ONE ENTRY
028A  20    EF              BRA     LTLP            ;TRY AGAIN
                                ;
028C  DE    00G     LCTGZ:  LDX     ZX,D            ;CLEAR R0 IF END OF STACK
028E  0F    00G             STX     R0,D
0290  39                    RTS
                                ;
0291  86    01      LCTGEX: LDA A   1,I             ;SET NZ
0293  39                    RTS
```

XXX LOCTGR LOCATE A TAG IN A RANGE STARTING AT R0

```
                                        .SBTTL  XXX LOCTGR      LOCATE A TAG IN A RANGE STARTING AT R0
                                        .GLOBL  LOCTGR
                                ;
                                ;       INPUTS
                                ;               LOW VALUE TAG IS IN ACC-A
                                ;               HIGH VALUE TAG IS IN ACC-B
                                ;               STACK START ADDR-1 IS IN R0
                                ;
                                ;       OUTPUTS
                                ;               R0 IS ZERO IF VALUE IN RANGE IS NOT FOUND
                                ;               R0 IS STACK PTR-1 FOR FIRST MATCH FOUND
                                ;
0294    97      00G     LOCTGR: STA A   R1,D            ;SAVE MIN VALID TAG VALUE
0296    D7      01G             STA B   R1+1,D  ;SAVE MAX VALID TAG VALUE
0298    DE      00G     LTRLP:  LDX     R0,D            ;GET CURRENT TAG PTR
029A    A6      01              LDA A   1,X             ;GET CURRENT TAG
029C    91      00G             CMP A   R1,D
029E    25      08              BCS     LTRFL           ;IF TOO SMALL BRANCH
02A0    91      01G             CMP A   R1+1,D
02A2    23      ED              BLS     LCTGEX          ;IF LE ALL DONE
02A4    81      00G             CMP A   EOSTG,I
02A6    27      E4              BEQ     LCTGZ           ;IF END OF STACK RETURN ZERO
02A8    BD      02AD'   LTRFL:  JSR     BACKUP          ;BACK OFF ONE ENTRY
02AB    20      EB              BRA     LTRLP
```

```
                                        .SBTTL  *** BACKUP      BACK UP ONE ENTRY ON THE STACK FROM R0
                                        .GLOBL  BACKUP
                                ;
                                ;       BACK UP ONE ENTRY ON STACK
                                ;
                                ;       R0 IS USED AS STACK PTR
                                ;
                                ;       IF AT END OF STACK THIS IS A NOP
                                ;
02AD  DE    00G     BACKUP: LDX     R0,D            ;START AT R0
02AF  A6    01              LDA A   1,X
02B1  81    00G             CMP A   EOSTG,I         ;MAKE SURE THIS IS SAFE
02B3  22    16              BHI     BCKERR          ;STACK IS DEAD
02B5  CE    02CE'           LDX     TAGTBL,I        ;GET ADDR OF TABLE
02B8  BD    0000G           JSR     L08X            ;GET LENGTH OF ENTRY
02BB  2A    04              BPL     BCKADD          ;IF SIGN BIT SET THIS IS VARIABLE LENGTH ENTRY
02BD  DE    00G             LDX     R0,D
02BF  E6    02              LDA B   2,X             ;LENGTH IS FIRST BYTE OF ENTRY
02C1  4F            BCKADD: CLR A                   ;CALC NEW STACK ADDR
02C2  DB    01G             ADD B   R0+1,D
02C4  99    00G             ADC A   R0,D
02C6  97    00G             STA A   R0,D            ;RETURN IT TO THE USER
02C8  D7    01G             STA B   R0+1,D
02CA  39                    RTS
                                ;
02CB  BD    0000G   BCKERR: JSR     SYSERR          ;BETTER LUCK NEXT TIME
                                ;
02CE  01            TAGTBL: .BYTE   1       ;NULLTG
02CF  05                    BYTE    5       ;PLOSTG
02D0  09                    .BYTE   9.      ;ESTG
02D1  03                    .BYTE   3       ;GOSTG
02D2  17                    BYTE    23      ;FORTG
02D3  03                    .BYTE   3       ;LISTTG
02D4  03                    .BYTE   3       ;PGMTG
02D5  03                    BYTE    3       ;IMXTG
02D6  05                    .BYTE   5       ;PNTSTG
02D7  05                    .BYTE   5       ;PSCTG
02D8  09                    BYTE    9       ;PNTNTG
02D9  09                    .BYTE   9.      ;PAETG
02DA  09                    .BYTE   9.      ;VALTG
02DB  03                    BYTE    3       ;LNNOTG
02DC  03                    .BYTE   3       ;ITM1TG
02DD  03                    .BYTE   3       ;ITM2TG
02DE  01                    BYTE    1       ;LBRKTG
02DF  01                    .BYTE   1       ;SEMITG
02E0  01                    .BYTE   1       ;ATSNTG
02E1  04                    BYTE    4       ;BAKSTG
02E2  01                    .BYTE   1       ;ALLTG
02E3  03                    .BYTE   3       ;RTRNTG
02E4  80                    BYTE    128     ;PRTTG
02E5  05                    .BYTE   5       ;CALLTG
02E6  03                    .BYTE   3       ;EOLTG
02E7  00                    BYTE    0       ;EOSTG
```

```
                                        .SBTTL  XXX NOP         NO-OPERATION
                                        .GLOBL  NOP
                                ;
                                ;       THIS ROUTINE ISUSED TO DO NOTHING
                                ;       IT IS USED TO IMPLEMENT NO OPERATION SYSTEM TOKENS
                                ;
02E8    39              NOP:    RTS
                                ;
                                        .SBTTL  XXX HALT        HALT LINE ROUTINE
                                        .GLOBL  HALTR,HALTA
                                ;
                                ;       THESE ROUTINES ARE USED TO STOP EXECUTION OF A LINE WITHOUT GETTING TO EOL.
                                ;
02E9    86      40      HALTR:  LDA A   ABRFLG,I        ;HALT LINE EXECUTION
02EB    9A      00G             ORA A   LCLFLG,D
02ED    97      00G             STA A   LCLFLG,D
02EF    39                      RTS
                                ;
02F0    8D      F7      HALTA:  BSR     HALTR
02F2    7E      0000G           JMP     DREXTQ
```

```
                                        .SBTTL  *** SETERR      SET ERROR CODE
                                        .GLOBL  SETERR,SETERA

                                ;
                                ;       SET ERROR CODE AND RTS ONE UP OR TAKE DREXTA
                                ;
02F5    CE      0306'   SETERR: LDX     EXIT,I
02F8    DF        01G           STX     DREXTA+1,D
02FA    30              SETERA: TSX
02FB    EE        00            LDX     0,X
02FD    A6        00            LDA A   0,X
02FF    97        00G           STA A   ERRCD,D
0301    32                      PUL A
0302    32                      PUL A
0303    7E      00C0G           JMP     DREXTA
                                ;
0306    39              EXIT:   RTS
                                ;
                0001'           .END
```

SYMBOL TABLE

```
ABRFLG=  0040        AFAIL =  0030        ALLOK =  0004        ALLTG = XXXXXX G     AMAT  =  0010
ARRAY =  0020        ASGCOD= XXXXXX G     ASTR  =  0020        ATSNTG= XXXXXX G     AEX   = XXXXXX G
A8X   = XXXXXX G     A9X   = XXXXXX G     BACKUP   02ADRG      BAKSTG= XXXXXX G     BCKADD   02C1R
BCKERR   02C8R       BMAT  =  0004        BSTR  =  0008        CALLTG= XXXXXX G     CDOPTR= XXXXXX G
CDSPTR= XXXXXX G     CLPTR = XXXXXX G     CLRARG   0004RG      CMAT  =  0001        CONCOD= XXXXXX G
CRCOD = XXXXXX G     CSTR  =  0002        CTKN  = XXXXXX G     DATCOU= XXXXXX G     DIMFLG=  0004
DISBLE= XXXXXX G     DISSRQ=  0008        DREXTA= XXXXXX G     DREXTB= XXXXXX G     ENABLE= XXXXXX G
EOECOD= XXXXXX G     EOETBL= XXXXXX G     EOLTG = XXXXXX G     EOSTG = XXXXXX G     EOUCOD= XXXXXX G
ERASGN= 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     ERNXFN= XXXXXX G
ERNXTA= XXXXXX G     EROFA = XXXXXX G     ERRCD = XXXXXX G     ERRCDB= XXXXXX G     ERSHOR= XXXXXX G
ERSTOP= XXXXXX G     ERUNOF= XXXXXX G     ERVAL = XXXXXX G     ERWSFL= XXXXXX G     ESTG  = XXXXXX G
EVLSUB=  000CRG      EXIT     0306R       EXTFLG=  0080        FIX1  = XXXXXX G     FNACOD= XXXXXX G
FNELG =  0010        FNTBL = XXXXXX G     FORTG = XXXXXX G     GETLMX   0231R       GETLN    0217RG
GETLNA   0212RG      GETLNX   0229R       GLBFLG= XXXXXX G     GLBLP    0259R       GLFAIL   026ER
GLFLP    0247R       GLHIT    0269R       GLSKIP   0235R       GOSTG = XXXXXX G     HALTA    02FORG
HSLTR    02E9RG      IMPCOD= XXXXXX G     IMXFLG=  0020        IMXIG = XXXXXX G     ITM1TG= XXXXXX G
ITM2TG= XXXXXX G     JMPAX = XXXXXX G     JMPX  = XXXXXX G     KEYFLG=  0010        KEYSTK= XXXXXX G
LBRKTG= XXXXXX G     LCIFLG= XXXXXX G     LCTGEX   0291R       LCTGZ    028CR       LDAX  = XXXXXX G
LDBX  = XXXXXX G     LDXX  = XXXXXX G     LISTTG= XXXXXX G     LITCOD= XXXXXX G     LNNOTG= XXXXXX G
LOCTG    0279RG      LOCTGR   0294RG      LSP   = XXXXXX G     LSTCOD= XXXXXX G     LTLP     027BR
LTRFL    0298R       LTRLP    0298R       MATSIZ   018ARG      MMICOD= XXXXXX G     MPLCOD= XXXXXX G
MSBY4    01D7R       MSCONT   01C2R       MSFLIP   01EAR       MSLP     01CAR       MSNORD   01D0R
MSSKP    01AAR       MULCOD= XXXXXX G     NCSFLG=  0001        NLPTR = XXXXXX G     NOP      02E8RG
NTAPTR=  0008        NTATTR=  0004        NTDIMS=  0009        NTDLEN=  0005        NTLINK=  0000
NTNOTE=  0002        NTPTR = XXXXXX G     NTSPTR=  0008        NTVAL =  0075        NTWCNT=  0007
NTWLEN=  0007        NTWROW=  0005        NULLTG= XXXXXX G     OBJATR=  CJ02        OBJBCK=  0003
OBJDAT=  0005        OBJLEN=  0000        ONSFLG=  0002        ONTBL = XXXXXX G     OPRADR= XXXXXX G
OVRFLG=  0002        PARITY = XXXXXX G    PARM  =  0008        PGMATR=  0002        PGMBP =  0005
PGMCD =  0009        PGMFP =  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     POPES    02DCRG      PRITG = XXXXXX G     PSCTG = XXXXXX G     PSHFPN= XXXXXX G
PSHRET= XXXXXX G     PULFPN= XXXXXX G     PUSHES   01F2RG      PUSHLP   01FBR       RFAIL =  00C0
RMAT  =  0040        RSTR  =  0080        RTRN  = XXXXXX G     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     SBP   = XXXXXX G     SCALER=  0040        SEMITG= XXXXXX G
SETARG   0000RG      SETERA   02FARG      SETERR   02F5RG      SIZCOD= XXXXXX G     STAX  = XXXXXX G
STPFLG=  0040        STPTR = XXXXXX G     STRING=  0010        SYSERR= XXXXXX G     TABPNT= XXXXXX G
TAGTBL   02CER       TAJMP    0029R       TALVS    00C3R       TALPA    0040R       TALPB    0071R
TAMAT    0084R       TAMAT2   0005R       TAMEND   00EAR       TAMFL    0104R       TAMLP    00F1R
TAMSKB   00EDR       TANTN    0036R       TANTS    0091R       TAOOPS   008AR       TAPAE    0063R
TATGVL   0058R       TAUNDF   0086R       TAVAL    00ECR       TRCFLG=  0020        TROKA    0137R
TROOPS   011AR       TRPNTN   0122R       TRSTR    0146R       TYPARG   0007RG      TYPRES   0107RG
T1    = XXXXXX G     T2    = XXXXXX G     T3    = XXXXXX G     UNDEF =  0080        VALERR=  0040
VALTG = XXXXXX G     ZX    = XXXXXX G
  ABS.    0000    00
          0307    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2478. WORDS
.SY:EVLSUB/CSDK:SEICLI,EVLSUB
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

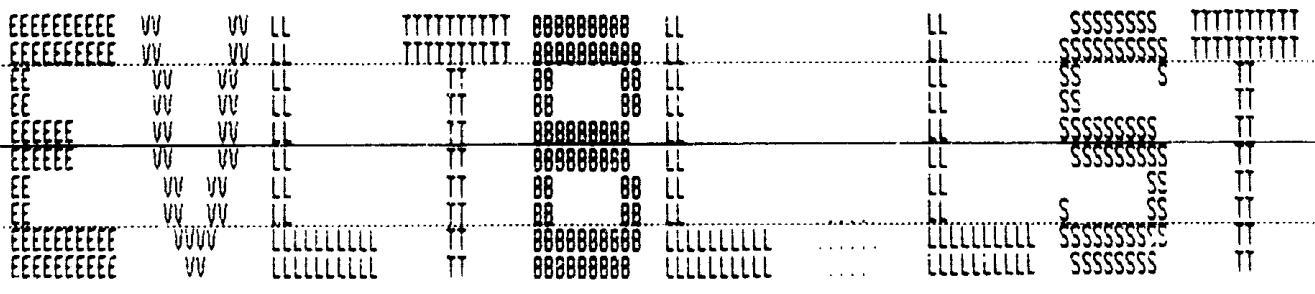
CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

```
TYPRES  8-2#    8-11#
UNDEF   4-8#    7-67    7-119   8-54    8-59
VALERR  4-29#   7-93    7-124   7-166
VALTG   5-16#   7-43    7-84
ZX      7-7#    12-68   13-22
```

CROSS REFERENCE TABLE (CREF V01-03)

PLRA 1-20#
PLRB 1-27# 10-7
SE1 1-3#



```
                                .TITLE  EVLTBL  TABLES FOR EVALUATOR
                                .IDENT  /BE6027/
                                ;
                                ;       THIS IS THE EVALUATOR DISPATCH TABLES.
                                ;       THERE ARE SEVERAL TYPES OF ENTRIES FOR TWO DISPATCHES FOR EACH
                                ;         TOKEN
                                ;
                                ;       THE FIRST OPERAND IS THE SYSTEM TOKEN CODE SYMBOL
                                ;       THE SECOND IS THE NAME OF THE ROUTINE TO CALL.  IF IT IS NOT GIVEN
                                ;         THE EVALUATOR WILL CALL THE CODE ID.
                                ;       THE THIRD ENTRY IS THE FIRST DISPATCH IF SPECIFIED.
                                ;         THE FIRST DISPATCH TEST FOR VALID DATA TYPES OR SETS UP THE
                                ;         I/O SYSTEM.  (THE I/O SYSTEM MAY NOT RETURN.)
                                ;       THE LAST ITEM IS SET TO HAVE THE EVALUATOR JUMP TO THE ROUTINE
                                ;         INSTEAD OF JSR THERE.
                                ;
0000'                           .CSECT  EVLTBL
                                ;
                                .GLOBL  OPRTBL          ;ARITH CLASS TEST PART OF TABLE
                                .GLOBL  CMDTBL          ;COMMAND SETUP SECTION
                                .GLOBL  BRKCOD          ;LAST ARITH FUNCTION CODE
                                .GLOBL  LSTCOD          ;LAST VALID CODE
                                .
                                .ENABL  HEX
                                ;
                                .MACRO  Z       ID,RTN,CLS,RTRN
                        ID'COD  =       TKN
                        TKN     =       TKN+1
                                .GLOBL  ID'COD
                                .IF     EQ,GEN
                                .MEXIT
                                .ENDC
                                .IF     B,RTN
                                .IF     B,RTRN
                                .WORD   ID
                                .IFF
                                .WORD   ID-32768
                                .ENDC
                                .GLOBL  ID
                                .IFF
                                .IF     B,RTRN
                                .WORD   RTN
                                .IFF
                                .WORD   RTN-32768
                                .ENDC
                                .GLOBL  RTN
                                .ENDC
                                .IF     B,CLS
                        WRK     =       0
                                .IFF
                        WRK     =       CLS
                                .ENDC
                                .IF     EQ,FLIP
                        HOLD    =       WRK
                        FLIP    =       1
                                .IFF
                                .CSECT  OCTBL
```

```
                                .BYTE   HOLD%16+WRK
                                .CSECT  EVLTBL
                        FLIP    =       0
                                .ENDC
                                .ENDM
                                ;
000A                            .RADIX  10
                                ;
0001                    C1      =       1       ;DEFINE FUNCTION CLASSES
0002                    C2      =       2
0003                    C3      =       3
0004                    C4      =       4
0005                    C5      =       5
0006                    C6      =       6
0007                    C7      =       7
0008                    C8      =       8
0009                    C9      =       9
000A                    CA      =       10
000B                    CB      =       11
000C                    CC      =       12
000D                    CD      =       13
000E                    CE      =       14
                                ;
0001                    SI      =       1       ;I/O PROC SETUP CALL
0002                    SF      =       2       ;FILES
0003                    SB      =       3       ;INC BRACKET COUNT AND PUSH
0004                    SD      =       4       ;DISPLAY FUNCTIONS
0005                    ST      =       5       ;MAG TAPE
0006                    SL      =       6       ;LISTING
0007                    PTG     =       7       ;PUSH STACK TAG ONLY ROUTINE
                                ;
                                ;
0001                    GEN     =       1       ;CONTROLS TABLE ENTRY GENERATION
0001                    TKN     =       1       ;TOKEN COUNTER
0001                    FLIP    =       1       ;CONTROLS GENERATION OF HALF BYTES
0000                    HOLD    =       0       ;HOLDING AREA FOR MACRO SET CALLS
```

```
                                  ;
  0000   0000G                    OPRTBL:  WORD   SYSERR          ; TOKEN OF ZERO IS INVALID
  0002                                     Z      MOV,MOVE,CA
  0004                                     Z      RMO,RMOVE,CA
  0006                                     Z      DRA,DRAW,CA
  0008                                     Z      RDR,RDRAW,CA
  000A                                     Z      SEG,,CB
  000C                                     Z      REP,,CB
  000E                                     Z      POS,,CC
  0010                                     Z      MIN,,C4
  0012                                     Z      MAX,,C4
  0014                                     Z      AND,,C4
  0016                                     Z      OR,,C4
  0018                                     Z      PLS,FPADD,C4    ; PLUS
  001A                                     Z      MNS,FPSUB,C4    ; MINUS
  001C                                     Z      MUL,FPMUL,C4    ; MULTIPLY
  001E                                     Z      DIV,FPDIV,C4    ; DIVISION
  0020                                     Z      UP,,C4          ; EXPONENTIATION
  0022                                     Z      EQU,EQ,C5       ; =
  0024                                     Z      NE,,C5          ; <>
  0026                                     Z      LT,,C5          ; <
  0028                                     Z      LTE,,C5         ; <=
  002A                                     Z      GTE,,C5         ; >=
  002C                                     Z      GT,,C5          ; >
  002E                                     Z      CAT,,C8         ; CATENATION
  0030                                     Z      MPY,,CE
  0032                                     Z      SUM,,CD,*
  0034                                     Z      TRN,,C9
  0036                                     Z      INV,,C9
  0038                                     Z      CHR,,C7
  003A                                     Z      STR,,C7
  003C                                     Z      ASC,,C6
  003E                                     Z      VAL,,C6
  0040                                     Z      LEN,,C6
  0042                                     Z      MPL,NOP,C2
  0044                                     Z      MMI,FPNEG,C2
  0046                                     Z      NOT,,C2
  0048                                     Z      FNA,FNEVL,C1    ; USER FUNCTIONS
         0000             GEN              =      0               ;*** NO TABLE ENTRIES FOR NOW ***
  004A                                     Z      FNB
  004A                                     Z      FNC
  004A                                     Z      FND
  GJ4A                                     Z      FNE
  004A                                     Z      FNF
  004A                                     Z      FNG
  004A                                     Z      FNH
  004A                                     Z      FNI
  004A                                     Z      FNJ
  004A                                     Z      FNK
  004A                                     Z      FNL
  004A                                     Z      FNM
  004A                                     Z      FNN
  004A                                     Z      FNO
  004A                                     Z      FNP
  004A                                     Z      FNQ
  004A                                     Z      FNR
  004A                                     Z      FNS
```

```
                                        Z       FNT
                                        Z       FNU
                                        Z       FNV
                                        Z       FNW
                                        Z       FNX
                                        Z       FNY
                                        Z       FNZ
                    0001        GEN     =       1               ;RESET TABLE GEN
                                        Z       TYP,TYPE,C1
                                        Z       ABS,,C2
                                        Z       SQR,,C2
                                        Z       SIG,,C2
                                        Z       RND,,C2
                                        Z       INT,,C2
                                        Z       LGN,,C2
                                        Z       LOG,,C2
                                        Z       EXP,ETOX,C2
                                        Z       ASN,,C2
                                        Z       ACS,,C2
                                        Z       ATN,,C2
                                        Z       TAN,TRIG,C2
                                        Z       COS,TRIG,C2
                                        Z       SIN,TRIG,C2
                                        Z       PT,,C1          ; PERCENT
                                        Z       IEC,,C1         ; AT
                                        Z       FIL,,C1         ; POUND
                                        Z       FOR,,C1,*
                                        Z       AS2,FNASGN,C1,*
                                        Z       GO,GOTO,C1,*    ; GO TO
                                        Z       GOS,GOSUB,C1,*  ; GOSUB
                                        Z       IF,,C3,*
                                ;
                    003C        BRKCOD  =       TKN-25          ;TABLE BREAK POINT
                                ;                               ASSUMES RELOCATION OF FNB TO FNZ INTO FNA
                                ;
                                CMDTBL: Z       RUN,,*
                                        Z       RES,RESTS,,*    ; RESTORE
                                        Z       REN,RENUMB
                                        Z       DEL,DELETE
                                        Z       APP,APPEND
                                        Z       LIS,LIST,SL
                                        Z       SAV,SAVE,SL
                                        Z       PRI,PRINT,SI
                                        Z       NEX,NEXT,,*
                                        Z       RET,RETURN,,*
                                        Z       STO,STOP
                                        Z       END
                                        Z       SET,NOP
                                        Z       INI,INIT
                                        Z       GIN
                                        Z       POL,POLL
                                        Z       LET,NOP
                                        Z       MAT,NOP
                                        Z       REA,READ,SI
                                        Z       WRI,WRITE,SI
                                        Z       INP,INPUT,SI
                                        Z       CAL,CALL
```

```
115  00A4  2  WIN,WINDOW
116  00A6  2  VIE,VIEW
117  00A8  2  AXI,AXIS
118  00AA  2  OLD,,SL
119  00AC  2  CRE,CREATE,SF
120  00AE  2  OPE,OPEN,SF
121  00B0  2  KIL,KILL
122  00B2  2  SEC,SECENC,SD
123  00B4  2  COP,COPY
124  00B6  2  ASS,ASSIGN,SF
125  00B8  2  UNI,UNIT,SF
126  00BA  2  FIN,FIND,ST
127  00BC  2  RBY,RBYTE
128  00BE  2  WBY,WBYTE
129  00C0  2  MAR,MARK,ST
130  00C2  2  TLI,TLIST
131  00C4  2  PAG,PAGE,SD
132  00C6  2  HOM,HOME,SD
133  00C8  2  DIR,,SF
134  00CA  2  CLO,CLOSE
135  00CC  2  WAI,WAIT
136  00CE  2  FUZ,SETFUZ
137  00D0  2  SCA,SCALE
138  00D2  2  ROT,ROTATE
139  00D4  2  CMD,,SF          ; I/O GENERAL COMMAND
140  00D6  2  CRO,CROSS
141  00D8  2  AS
142  00DA  2  TO,NOP
143  00DC  2  TO2,NOP
144  00DE  2  STE,NOP          ; STEP
145  00E0  2  THE,NOP          ; THEN
146  00E2  2  REL,RELBL,SF     ; RELABEL
147  00E4  2  OF,,,%
148  00E6  2  ASG,,,%
149  00E8  2  COL,SYSERR       ; COLON
150  00EA  2  SEM,SYSERR       ; SEMI-COLON
151  00EC  2  RPN              ; RIGHT PAREN
152  00EE  2  COM,SYSERR       ; COMMA
153  00F0  2  USI,USING,,%
154  00F2  2  LPN,SYSERR       ; LEFT PAREN
155  00E4  2  LBK,LBRKTG,SB    ; LEFT BRACKET
156  00F6  2  CR,EOL           ; CARRIAGE RETURN
157  00F8  2  GRA,GRAD
158  00FA  2  DEG
159  00FC  2  RAD
160  00FE  2  TRA,TRACE
161  0100  2  NOR,NORMAL
162  0102  2  KEY
163  0104  2  NOK,NOKEY
164  0106  2  CAS,CASE
165  0108  2  NOC,NOCASE
166  010A  2  EOF,,,%
167  010C  2  PI
168  010E  2  SPA,SPACE,,%
169  0110  2  MEM,MEMORY,,%
170  0112  2  DET
171  0114  2  IMA,HALTR         ; IMAGE
```

```
0116                    Z       REM,HALTR        ; REMARK
0118                    Z       DAT,HALTR        ; DATA
011A                    Z       DIM              ; DIMENSION
011C                    Z       DEF              ; DEFINE
011E                    Z       OFF
0120                    Z       ON
0122                    Z       SIZ,ONUNIT,,*    ; SIZE
0124                    Z       FUL,FLUNIT       ; FULL
0126                    Z       SRQ,ONUNIT,,*
0128                    Z       EOI,ONUNIT,,*
012A                    Z       ALL,ALLITG,PTG
012C                    Z       SM2,SEMITG,PTG   ; I DON'T KNOW
012E                    Z       NUM,PTRNTN,,*    ; NAME TABLE NUMERIC
0130                    Z       STG,PTRNTS,,*    ; NAME TABLE STRING
0132                    Z       ITG,INTCN,,*     ; INTEGER INLINE
0134                    Z       LIN,LINCN,,*     ; LINE NUMBER
0136                    Z       PRM,ENPARM,,*    ; PARAMETER
0138                    Z       LIT,LITCN,,*     ; LITERAL CONSTANT
013A                    Z       CON,DATACN,,*    ; CONSTANT

        0086    LSTCOD  =       TKN-1            ; LAST VALID CODE
                ;
                        .IF     NE,FLIP
                        .CSECT  OCTBL
                        .BYTE   HOLD*16          ; FORCE LAST ENTRY TO GEN
                        .ENDC
                ;
        0001'           END
```

SYMBOL TABLE

```
ABS   = XXXXXX G      ABSCOD= 003F G      ACS   = XXXXXX G      ACSCOD= 0048 G      ALLCOD= 00AE G
ALLTG = XXXXXX G      AND   = XXXXXX G      ANDCOD= 0009 G      APPCOD= 0059 G      APPEND= XXXXXX G
AS    = XXXXXX G      ASC   = XXXXXX G      ASCCOD= 001E G      ASCOD = 0085 G      ASG   = XXXXXX G
ASGCOD= 008C G      ASN   = XXXXXX G      ASNCOD= 0047 G      ASSCOD= 0074 G      ASSIGN= XXXXXX G
AS2COD= 0051 G      ATN   = XXXXXX G      ATNCOD= 0049 G      AXICOD= 0060 G      AXIS  = XXXXXX G
BRKCOD= 003C G      CA    = 000A          CALCOD= 006A G      CALL  = XXXXXX G      CASCOD= 009C G
CASE  = XXXXXX G      CAT   = XXXXXX G      CATCOD= 0017 G      CB    = 0038          CC    = 000C
CD    = 0000          CE    = 000E          CHR   = XXXXXX G      CHRCOD= 0A1C G      CLOCOD= 002E G
CLOSE = XXXXXX G      CMD   = XXXXXX G      CMDCOD= 0083 G      CMDTBL  0078RG    02  COLCOD= 0080 G
COMCOD= 0090 G      CONCOD= 0086 G      COPCOD= 0073 G      COPY  = XXXXXX G      COSCOD= 0048 G
CRCOD = 0094 G      CREATE= XXXXXX G      CRECOD= 006F G      CROCOD= 008A G      CROSS = XXXXXX G
C1    = 0001          C2    = 0002          C3    = 0003          C4    = 0004          C5    = 0005
C6    = 0006          C7    = 0007          C8    = 0008          C9    = 0009          DATACN= XXXXXX G
DATCOD= 0095 G      DEF   = XXXXXX G      DEFCOD= 0097 G      DEG   = XXXXXX G      DEGCOD= 0096 G
DELCOD= 0058 G      DELETE= XXXXXX G      DET   = XXXXXX G      DETCOD= 00A2 G      DIM   = XXXXXX G
DIMCOD= 00A6 G      DIR   = XXXXXX G      DIRCOD= 0070 G      DIVCOD= 000F G      DRACOD= 0003 G
DRAW  = XXXXXX G      END   = XXXXXX G      ENDCOD= 0060 G      EOF   = XXXXXX G      EOFCOD= 009E G
EOICOD= 00AD G      EOL   = XXXXXX G      EQ    = XXXXXX G      EQUCOD= 0011 G      ETOX  = XXXXXX G
EXPCOD= 0046 G      FIL   = XXXXXX G      FILCOD= 004F G      FINCOD= 0076 G      FIND  = XXXXXX G
FLIP  = 0000          FLUNIT= XXXXXX G      FNACOD= 0024 G      FNCSGN= XXXXXX G      FNBCOD= 0025 G
FNCCOD= 0026 G      FNDCOD= 0027 G      FNECOD= 0028 G      FNEVL = XXXXXX G      FNFCOD= 0029 G
FNGCOD= 002A G      FNHCOD= 002B G      FNICOD= 002C G      FNJCOD= 002D G      FNKCOD= 002E G
FNLCOD= 002F G      FNMCOD= 0030 G      FNNCOD= 0031 G      FNOCOD= 0032 G      FNPARM= XXXXXX G
FNPCOD= 0033 G      FNQCOD= 0034 G      FNRCOD= 0035 G      FNSCOD= 0036 G      FNTCOD= 0037 G
FNUCOD= 0038 G      FNVCOD= 0039 G      FNWCOD= 003A G      FNXCOD= 003B G      FNYCOD= 003C G
FNZCOD= 003D G      FOR   = XXXXXX G      FORCOD= 0050 G      FPADD = XXXXXX G      FPDIV = XXXXXX G
FPML  = XXXXXX G      FPNEG = XXXXXX G      FPSUB = XXXXXX G      FULCOD= 0048 G      FUZCOD= 0080 G
GEN   = 0001          GIN   = XXXXXX G      GINCOD= 0063 G      GOCOD = 0052 G      GOSCOD= 0053 G
GOSUB = XXXXXX G      GOTO  = XXXXXX G      GRACOD= 0095 G      GRAD  = XXXXXX G      GT    = XXXXXX G
GTCOD = 0016 G      GTE   = XXXXXX G      GTECOD= 0015 G      HALTR = XXXXXX G      HOLD  = 0000
HOMCOD= 007C G      HOME  = XXXXXX G      IEC   = XXXXXX G      IECCOD= 004E G      IF    = XXXXXX G
IFCOD = 0054 G      IMACOD= 0093 G      INICOD= 0062 G      INIT  = XXXXXX G      INPCOD= 0069 G
]]]]]
INPUT = XXXXXX G      INT   = XXXXXX G      INTCN = XXXXXX G      INTCOD= 0043 G      INV   = XXXXXX G
INVCOD= 001B G      ITGCOD= 0082 G      KEY   = XXXXXX G      KEYCOD= 009A G      KILCOD= 0071 G
KILL  = XXXXXX G      LBKCOD= 0093 G      LBRKTG= XXXXXX G      LEN   = XXXXXX G      LENCOD= 0020 G
LETCOD= 0065 G      LGN   = XXXXXX G      LGNCOD= 0044 G      LINCN = XXXXXX G      LINCOD= 0083 G
LISCOD= 0068 G      LIST  = XXXXXX G      LITCN = XXXXXX G      LITCOD= 0085 G      LOG   = XXXXXX G
LOGCOD= 0045 G      LPNCOD= 0092 G      LSTCOD= 0086 G      LT    = XXXXXX G      LTCOD = 0013 G
LTE   = XXXXXX G      LTECOD= 0014 G      MARCOD= 0079 G      MARK  = XXXXXX G      MATCOD= 0066 G
MAX   = XXXXXX G      MAXCOD= 0009 G      MEMCOD= 00A1 G      MEMORY= XXXXXX G      MIN   = XXXXXX G
MINCOD= 0008 G      MMICOD= 0022 G      MNSCOD= 000D G      MOVCOD= 0001 G      MOVE  = XXXXXX G
MPLCOD= 0021 G      MPY   = XXXXXX G      MPYCOD= 0018 G      MULCOD= 000E G      NE    = XXXXXX G
NECOD = 0012 G      NEXCOD= 005D G      NEXT  = XXXXXX G      NOCASE= XXXXXX G      NOCCOD= 0090 G
NORCOD= 0098 G      NOKEY = XXXXXX G      NOP   = XXXXXX G      NORCOD= 0099 G      NORMAL= XXXXXX G
NOT   = XXXXXX G      NOTCOD= 0023 G      NUMCOD= 0080 G      OF    = XXXXXX G      OFCOD = 0088 G
OFF   = XXXXXX G      OFFCOD= 00A8 G      OLD   = XXXXXX G      OLDCOD= 006E G      ON    = XXXXXX G
ONCOD = 0089 G      ONUNIT= XXXXXX G      OPECOD= 0070 G      OPEN  = XXXXXX G      OPRTBL  0000RG    02
OR    = XXXXXX G      ORCOD = 0008 G      PAGCOD= 0078 G      PAGE  = XXXXXX G      PI    = XXXXXX G
PICOD = 009F G      PLSCOD= 000C G      POLCOD= 0064 G      POLL  = XXXXXX G      POS   = XXXXXX G
POSCOD= 0007 G      PRICOD= 005C G      PRINT = XXXXXX G      PRMCOD= 0084 G      R     = XXXXXX G
PTCOD = 004D G      PTG   = 0007          PTRNTN= XXXXXX G      PTRNTS= XXXXXX G      RAD   = XXXXXX G
RADCOD= 0097 G      RBYCOD= 0077 G      RBYTE = XXXXXX G      RDROW = XXXXXX G      RDRCOD= 0004 G
REACOD= 0067 G      READ  = XXXXXX G      RELBL = XXXXXX G      RELCOD= 008A G      REMCOD= 00A4 G
RENCOD= 0057 G      RENUMB= XXXXXX G      REP   = XXXXXX G      REPCOD= 0006 G      RESCOD= 0056 G
RESTS = XXXXXX G      RETCOD= 005E G      RETURN= XXXXXX G      RMOCOD= 0002 G      RMOVE = XXXXXX G
RND   = XXXXXX G      RNDCOD= 0042 G      ROTATE= XXXXXX G      ROTCOD= 0082 G      RPN   = XXXXXX G
RPNCOD= 008F G      RUN   = XXXXXX G      RUNCOD= 0055 G      SAVCOD= 0053 G      SAVE  = XXXXXX G
```

```
EVLTBL TABLES FOR EVALUATOR    RT-11 MMAC  VM02-10   14-OCT-76 01:31:32 PAGE 2+
SYMBOL TABLE

SB    =  0003        SCACOD=  0081 G      SCALE = XXXXXX G      SD    =  0004        SECCOD=  0072 G
SECFNC= XXXXXX G     SEG   = XXXXXX G     SEGCOD=  0005 G     SEMCOD=  008E G     SEMLTG= XXXXXX G
SETCOD=  0061 G      SETFUZ= XXXXXX G     SF    =  0002       SI    =  0001       SIG   = XXXXXX G
SIGCOD=  0041 G      SINCOD=  004C G      SIZCOD=  00AA G     SL    =  0006       SM2COD=  00AF G
SPACE = XXXXXX G     SPACOD=  0080 G      SQR   = XXXXXX G     SQRCOD=  0040 G     SROCOD=  00BC G
ST    =  0005        STECOD=  0088 G      STGCOD=  0081 G     STOCOD=  005F G     STOP  = XXXXXX G
STR   = XXXXXX G     STRCOD=  001D G      SUM   = XXXXXX G     SUMCOD=  0019 G     SYSERR= XXXXXX G
TANCOD=  004A G      THECOD=  0089 G      TKN   =  0087       TLICOD=  0078 G     TLIST = XXXXXX G
TOCOD =  0086 G      TO2COD=  0087 G      TRACE = XXXXXX G     TRACOD=  0098 G     TRIG  = XXXXXX G
TRN   = XXXXXX G     TRNCOD=  001A G      TYPCOD=  003E G     TYPE  = XXXXXX G     UNICOD=  0075 G
UNIT  = XXXXXX G     UP    = XXXXXX G     UPCOD =  0010 G     USICOD=  0091 G     USING = XXXXXX G
VAL   = XXXXXX G     VALCOD=  001F G      VIECOD=  006C G     VIEW  = XXXXXX G     WAICOD=  007F G
WAIT  = XXXXXX G     WBYCOD=  0078 G      WBYTE = XXXXXX G     WINCOD=  0068 G     WINDOW= XXXXXX G
WRICOD=  0068 G      WRITE = XXXXXX G     WRK   =  0000

. ABS.    0000    00
          0000    01
EVLTBL    013C    02
OCTBL     004F    03
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 1834. WORDS
.SY:EVLTBL/C/OK1:SE!CL!.EVLTBL
```

CROSS REFERENCE TABLE (CREF V01-03)

ABS	2-67	2-67#													
ABSCOD	2-67#	2-67#													
ACS	2-76	2-76#													
ACSCOD	2-76#	2-76#													
ALLCOD	2-182#	2-182#													
ALLTG	2-182	2-182#													
AND	2-12	2-12#													
ANDCOD	2-12#	2-12#													
APPCOD	2-97#	2-97#													
APPEND	2-97	2-97#													
AS	2-141	2-141#													
AS2COD	2-85#	2-85#													
ASC	2-32	2-32#													
ASCCOD	2-32#	2-32#													
ASCOD	2-141#	2-141#													
ASG	2-148	2-148#													
ASGCOD	2-148#	2-148#													
ASN	2-75	2-75#													
ASNCOD	2-75#	2-75#													
ASSCOD	2-124#	2-124#													
ASSIGN	2-124	2-124#													
ATN	2-77	2-77#													
ATNCOD	2-77#	2-77#													
AXICOD	2-117#	2-117#													
AXIS	2-117	2-117#													
BRKCOD	1-36#	2-90#													
C1	1-81#	2-38	2-66	2-81	2-82	2-83	2-84	2-85	2-86	2-87					
C2	1-82#	2-35	2-36	2-37	2-67	2-68	2-69	2-70	2-71	2-72	2-73	2-74	2-75	2-76	
	2-77	2-78	2-79	2-80											
C3	1-83#	2-88													
C4	1-84#	2-10	2-11	2-12	2-13	2-14	2-15	2-16	2-17	2-18					
C5	1-85#	2-19	2-20	2-21	2-22	2-23	2-24								
C6	1-86#	2-32	2-33	2-34											
C7	1-87#	2-30	2-31												
C8	1-88#	2-25													
C9	1-89#	2-28	2-29												
CA	1-90#	2-3	2-4	2-5	2-6										
CALCOD	2-114#	2-114#													
CALL	2-114	2-114#													
CASCOD	2-164#	2-164#													
CASE	2-164	2-164#													
CAT	2-25	2-25#													
CATCOD	2-25#	2-25#													
CB	1-91#	2-7	2-8												
CC	1-92#	2-9													
CD	1-93#	2-27													
CE	1-94#	2-26													
CHR	2-30	2-30#													
CHRCOD	2-30#	2-30#													
CLOCOD	2-134#	2-134#													
CLOSE	2-134	2-134#													
CMD	2-139	2-139#													
CMDCOD	2-139#	2-139#													
CMDTBL	1-35#	2-93#													
COLCOD	2-149#	2-149#													
COMCOD	2-152#	2-152#													

CROSS REFERENCE TABLE (CREF V01-03.)

CONCOD	2-190#	2-190#
COPCOD	2-123#	2-123#
COPY	2-123	2-123#
COSCOD	2-79#	2-79#
CRCOD	2-156#	2-156#
CREATE	2-119	2-119#
CRECOD	2-119#	2-119#
CROCOD	2-140#	2-140#
CROSS	2-140	2-140#
DATACN	2-190	2-190#
DATCOD	2-173#	2-173#
DEF	2-175	2-175#
DEFCOD	2-175#	2-175#
DEG	2-158	2-158#
DEGCOD	2-158#	2-158#
DELCOD	2-96#	2-96#
DELETE	2-96	2-96#
DET	2-170	2-170#
DETCOD	2-170#	2-170#
DIM	2-174	2-174#
DIMCOD	2-174#	2-174#
DIR	2-133	2-133#
DIRCOD	2-133#	2-133#
DIVCOD	2-17#	2-17#
DRACOD	2-5#	2-5#
DRAW	2-5	2-5#
END	2-104	2-104#
ENDCOD	2-104#	2-104#
EOF	2-166	2-166#
EOFCOD	2-166#	2-166#
EQICOD	2-181#	2-181#
EOL	2-156	2-156#
EQ	2-19	2-19#
EQUCOD	2-19#	2-19#
ETOX	2-74	2-74#
EXPCOD	2-74#	2-74#
FIL	2-83	2-83#
FILCOD	2-83#	2-83#
FINCOD	2-126#	2-126#
FIND	2-126	2-126#

FLIP	1-107#	2-3	2-3#	2-4	2-4#	2-5	2-5#	2-6	2-6#	2-7	2-7#	2-8	2-8#	2-9
	2-9#	2-10	2-10#	2-11	2-11#	2-12	2-12#	2-13	2-13#	2-14	2-14#	2-15	2-15#	2-16
	2-16#	2-17	2-17#	2-18	2-18#	2-19	2-19#	2-20	2-20#	2-21	2-21#	2-22	2-22#	2-23
	2-23#	2-24	2-24#	2-25	2-25#	2-26	2-26#	2-27	2-27#	2-28	2-28#	2-29	2-29#	2-30
	2-30#	2-31	2-31#	2-32	2-32#	2-33	2-33#	2-34	2-34#	2-35	2-35#	2-36	2-36#	2-37
	2-37#	2-38	2-38#	2-40	2-41	2-42	2-43	2-44	2-45	2-46	2-47	2-48	2-49	2-50
	2-51	2-52	2-53	2-54	2-55	2-56	2-57	2-58	2-59	2-60	2-61	2-62	2-63	2-64
	2-66	2-66#	2-67	2-67#	2-68	2-68#	2-69	2-69#	2-70	2-70#	2-71	2-71#	2-72	2-72#
	2-73	2-73#	2-74	2-74#	2-75	2-75#	2-76	2-76#	2-77	2-77#	2-78	2-78#	2-79	2-79#
	2-80	2-80#	2-81	2-81#	2-82	2-82#	2-83	2-83#	2-84	2-84#	2-85	2-85#	2-86	2-86#
	2-87	2-87#	2-88	2-88#	2-93	2-93#	2-94	2-94#	2-95	2-95#	2-96	2-96#	2-97	2-97#
	2-98	2-98#	2-99	2-99#	2-100	2-100#	2-101	2-101#	2-102	2-102#	2-103	2-103#	2-104	2-104#
	2-105	2-105#	2-106	2-106#	2-107	2-107#	2-108	2-108#	2-109	2-109#	2-110	2-110#	2-111	2-111#
	2-112	2-112#	2-113	2-113#	2-114	2-114#	2-115	2-115#	2-116	2-116#	2-117	2-117#	2-118	2-118#
	2-119	2-119#	2-120	2-120#	2-121	2-121#	2-122	2-122#	2-123	2-123#	2-124	2-124#	2-125	2-125#
	2-126	2-126#	2-127	2-127#	2-128	2-128#	2-129	2-129#	2-130	2-130#	2-131	2-131#	2-132	2-132#
	2-133	2-133#	2-134	2-134#	2-135	2-135#	2-136	2-136#	2-137	2-137#	2-138	2-138#	2-139	2-139#

CROSS REFERENCE TABLE (CREF V01-03)

	2-140	2-140#	2-141	2-141#	2-142	2-142#	2-143	2-143#	2-144	2-144#	2-145	2-145#	2-146	2-146#
	2-147	2-147#	2-148	2-148#	2-149	2-149#	2-150	2-150#	2-151	2-151#	2-152	2-152#	2-153	2-153#
	2-154	2-154#	2-155	2-155#	2-156	2-156#	2-157	2-157#	2-158	2-158#	2-159	2-159#	2-160	2-160#
	2-161	2-161#	2-162	2-162#	2-163	2-163#	2-164	2-164#	2-165	2-165#	2-166	2-166#	2-167	2-167#
	2-168	2-168#	2-169	2-169#	2-170	2-170#	2-171	2-171#	2-172	2-172#	2-173	2-173#	2-174	2-174#
	2-175	2-175#	2-176	2-176#	2-177	2-177#	2-178	2-178#	2-179	2-179#	2-180	2-180#	2-181	2-181#
	2-182	2-182#	2-183	2-183#	2-184	2-184#	2-185	2-186#	2-186	2-186#	2-187	2-187#	2-188	2-188#
	2-189	2-189#	2-190	2-190#	2-194									
FLUNIT	2-179	2-179#												
FNACOD	2-38#	2-38#												
FNASGN	2-85	2-85#												
FNBCOD	2-40#	2-40#												
FNCCOD	2-41#	2-41#												
FNDCOD	2-42#	2-42#												
FNECOD	2-43#	2-43#												
FNEWL	2-38	2-38#												
FNFCOD	2-44#	2-44#												
FNGCOD	2-45#	2-45#												
FNHCOD	2-46#	2-46#												
FNICOD	2-47#	2-47#												
FNJCOD	2-48#	2-48#												
FNKCOD	2-49#	2-49#												
FNLCOD	2-50#	2-50#												
FNMCOD	2-51#	2-51#												
FNNCOD	2-52#	2-52#												
FNOCOD	2-53#	2-53#												
FNPARM	2-188	2-188#												
FNPCOD	2-54#	2-54#												
FNQCOD	2-55#	2-55#												
FNRCOD	2-56#	2-56#												
FNSCOD	2-57#	2-57#												
FNTCOD	2-58#	2-58#												
FNUCOD	2-59#	2-59#												
FNVCOD	2-60#	2-60#												
FNWCOD	2-61#	2-61#												
FNXCOD	2-62#	2-62#												
FNYCOD	2-63#	2-63#												
FNZCOD	2-64#	2-64#												
FOR	2-84	2-84#												
FORCOD	2-84#	2-84#												
FPADD	2-14	2-14#												
FPDIV	2-17	2-17#												
FPMUL	2-16	2-16#												
FPNEG	2-36	2-36#												
FPSUB	2-15	2-15#												
FULCOD	2-179#	2-179#												
FUZCOD	2-136#	2-136#												
GEN	1-105#	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10	2-11	2-12	2-13	2-14	2-15
	2-16	2-17	2-18	2-19	2-20	2-21	2-22	2-23	2-24	2-25	2-26	2-27	2-28	2-29
	2-30	2-31	2-32	2-33	2-34	2-35	2-36	2-37	2-38	2-39#	2-40	2-41	2-42	2-43
	2-44	2-45	2-46	2-47	2-48	2-49	2-50	2-51	2-52	2-53	2-54	2-55	2-56	2-57
	2-58	2-59	2-60	2-61	2-62	2-63	2-64	2-65#	2-66	2-67	2-68	2-69	2-70	2-71
	2-72	2-73	2-74	2-75	2-76	2-77	2-78	2-79	2-80	2-81	2-82	2-83	2-84	2-85
	2-86	2-87	2-88	2-93	2-94	2-95	2-96	2-97	2-98	2-99	2-100	2-101	2-102	2-103
	2-104	2-105	2-106	2-107	2-108	2-109	2-110	2-111	2-112	2-113	2-114	2-115	2-116	2-117
	2-118	2-119	2-120	2-121	2-122	2-123	2-124	2-125	2-126	2-127	2-128	2-129	2-130	2-131
	2-132	2-133	2-134	2-135	2-136	2-137	2-138	2-139	2-140	2-141	2-142	2-143	2-144	2-145

CROSS REFERENCE TABLE (CREF V01-03.)

	2-146	2-147	2-148	2-149	2-150	2-151	2-152	2-153	2-154	2-155	2-156	2-157	2-158	2-159
	2-160	2-161	2-162	2-163	2-164	2-165	2-166	2-167	2-168	2-169	2-170	2-171	2-172	2-173
	2-174	2-175	2-176	2-177	2-178	2-179	2-180	2-181	2-182	2-183	2-184	2-185	2-186	2-187
	2-188	2-189	2-190											
GIN	2-107	2-107#												
GINCOD	2-107#	2-107#												
GOCOD	2-86#	2-86#												
GOSCOD	2-87#	2-87#												
GOSUB	2-87	2-87#												
GOTO	2-86	2-86#												
GRACOD	2-157#	2-157#												
GRAD	2-157	2-157#												
GT	2-24	2-24#												
GTCOD	2-24#	2-24#												
GTE	2-23	2-23#												
GTECOD	2-23#	2-23#												
HALTR	2-171	2-171#	2-172	2-172#	2-173	2-173#								
HOLD	1-108#	2-3	2-4#	2-5	2-6#	2-7	2-8#	2-9	2-10#	2-11	2-12#	2-13	2-14#	2-15
	2-16#	2-17	2-18#	2-19	2-20#	2-21	2-22#	2-23	2-24#	2-25	2-26#	2-27	2-28#	2-29
	2-30#	2-31	2-32#	2-33	2-34#	2-35	2-36#	2-37	2-38#	2-66	2-67#	2-68	2-69#	2-70
	2-71#	2-72	2-73#	2-74	2-75#	2-76	2-77#	2-78	2-79#	2-80	2-81#	2-82	2-83#	2-84
	2-85#	2-86	2-87#	2-88	2-93#	2-94	2-95#	2-96	2-97#	2-98	2-99#	2-100	2-101#	2-102
	2-103#	2-104	2-105#	2-106	2-107#	2-108	2-109#	2-110	2-111#	2-112	2-113#	2-114	2-115#	2-116
	2-117#	2-118	2-119#	2-120	2-121#	2-122	2-123#	2-124	2-125#	2-126	2-127#	2-128	2-129#	2-130
	2-131#	2-132	2-133#	2-134	2-135#	2-136	2-137#	2-138	2-139#	2-140	2-141#	2-142	2-143#	2-144
	2-145#	2-146	2-147#	2-148	2-149#	2-150	2-151#	2-152	2-153#	2-154	2-155#	2-156	2-157#	2-158
	2-159#	2-160	2-161#	2-162	2-163#	2-164	2-165#	2-166	2-167#	2-168	2-169#	2-170	2-171#	2-172
	2-173#	2-174	2-175#	2-176	2-177#	2-178	2-179#	2-180	2-181#	2-182	2-183#	2-184	2-185#	2-186
	2-187#	2-188	2-189#	2-190										
HOMCOD	2-132#	2-132#												
HOME	2-132	2-132#												
IEC	2-82	2-82#												
IECCOD	2-82#	2-82#												
IF	2-88	2-88#												
IFCOD	2-88#	2-88#												
IMACOD	2-171#	2-171#												
INICOD	2-106#	2-106#												
INIT	2-106	2-106#												
INPCOD	2-113#	2-113#												
INPUT	2-113	2-113#												
INT	2-71	2-71#												
INTCN	2-186	2-186#												
INTCOD	2-71#	2-71#												
INV	2-29	2-29#												
INVCOD	2-29#	2-29#												
ITGCOD	2-186#	2-186#												
KEY	2-162	2-162#												
KEYCOD	2-162#	2-162#												
KILCOD	2-121#	2-121#												
KILL	2-121	2-121#												
LBKCOD	2-155#	2-155#												
LBRKTG	2-155	2-155#												
LEN	2-34	2-34#												
LENCOD	2-34#	2-34#												
LETCOD	2-109#	2-109#												
LGN	2-72	2-72#												
LGNCOD	2-72#	2-72#												

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

OPRTBL	1-34#	2-2#	
OR	2-13	2-13#	
ORCOD	2-13#	2-13#	
PAGCOD	2-131#	2-131#	
PAGE	2-131	2-131#	
PI	2-167	2-167#	
PICOD	2-167#	2-167#	
PLSCOD	2-14#	2-14#	
POLCOD	2-108#	2-108#	
POLL	2-108	2-108#	
POS	2-9	2-9#	
POSCOD	2-9#	2-9#	
PRICOD	2-100#	2-100#	
PRINT	2-100	2-100#	
PRMCOD	2-188#	2-188#	
PT	2-81	2-81#	
PTCOD	2-81#	2-81#	
PTG	1-102#	2-182	2-183
PTRNIN	2-184	2-184#	
PTRNTS	2-185	2-185#	
RAD	2-159	2-159#	
RADCOD	2-159#	2-159#	
RBYCOD	2-127#	2-127#	
RBYTE	2-127	2-127#	
RDRAW	2-6	2-6#	
RDRCOD	2-6#	2-6#	
REACOD	2-111#	2-111#	
READ	2-111	2-111#	
RELBL	2-146	2-146#	
RELCOD	2-146#	2-146#	
REMCOD	2-172#	2-172#	
RENCOD	2-95#	2-95#	
RENUMB	2-95	2-95#	
REP	2-8	2-8#	
REPCOD	2-8#	2-8#	
RESCOD	2-94#	2-94#	
RESTS	2-94	2-94#	
RETCOD	2-102#	2-102#	
RETURN	2-102	2-102#	
RMOCOD	2-4#	2-4#	
RMOVE	2-4	2-4#	
RND	2-70	2-70#	
RNDCOD	2-70#	2-70#	
ROTATE	2-138	2-138#	
ROTCOD	2-138#	2-138#	
RPN	2-151	2-151#	
RPNCOD	2-151#	2-151#	
RUN	2-93	2-93#	
RUNCOD	2-93#	2-93#	
SAVCOD	2-99#	2-99#	
SAVE	2-99	2-99#	
SB	1-98#	2-155	
SCACOD	2-137#	2-137#	
SCALE	2-137	2-137#	
SD	1-99#	2-122	2-131 2-132
SECCOD	2-122#	2-122#	
SECFNC	2-122	2-122#	

CROSS REFERENCE TABLE (CREF V01-03.)

SEG	2-7	2-7#												
SEGCOD	2-7#	2-7#												
SEMCOD	2-150#	2-150#												
SEMLIG	2-183	2-183#												
SETCOD	2-105#	2-105#												
SETFUZ	2-136	2-136#												
SF	1-97#	2-119	2-120	2-124	2-125	2-133	2-139	2-146						
SI	1-96#	2-100	2-111	2-112	2-113									
SIG	2-69	2-69#												
SIGCOD	2-69#	2-69#												
SINCOD	2-80#	2-80#												
SIZCOD	2-178#	2-178#												
SL	1-101#	2-98	2-99	2-118										
SM2COD	2-183#	2-183#												
SPACE	2-168	2-168#												
SPACOD	2-168#	2-168#												
SQR	2-68	2-68#												
SQRCOD	2-68#	2-68#												
SROCOD	2-180#	2-180#												
ST	1-100#	2-126	2-129											
STECOD	2-144#	2-144#												
STGCOD	2-185#	2-185#												
STOCOD	2-103#	2-103#												
STOP	2-103	2-103#												
STR	2-31	2-31#												
STRCOD	2-31#	2-31#												
SUM	2-27	2-27#												
SUMCOD	2-27#	2-27#												
SYSERR	2-2	2-149	2-149#	2-150	2-150#	2-152	2-152#	2-154	2-154#					
TANCOD	2-78#	2-78#												
THECOD	2-145#	2-145#												
TKN	1-106#	2-3	2-3	2-3#	2-4	2-4	2-4#	2-5	2-5	2-5#	2-6	2-6	2-6#	2-7
	2-7	2-7#	2-8	2-8	2-8#	2-9	2-9	2-9#	2-10	2-10	2-10#	2-11	2-11	2-11#
	2-12	2-12	2-12#	2-13	2-13	2-13#	2-14	2-14	2-14#	2-15	2-15	2-15#	2-16	2-16
	2-16#	2-17	2-17	2-17#	2-18	2-18	2-18#	2-19	2-19	2-19#	2-20	2-20	2-20#	2-21
	2-21	2-21#	2-22	2-22	2-22#	2-23	2-23	2-23#	2-24	2-24	2-24#	2-25	2-25	2-25#
	2-26	2-26	2-26#	2-27	2-27	2-27#	2-28	2-28	2-28#	2-29	2-29	2-29#	2-30	2-30
	2-30#	2-31	2-31	2-31#	2-32	2-32	2-32#	2-33	2-33	2-33#	2-34	2-34	2-34#	2-35
	2-35	2-35#	2-36	2-36	2-36#	2-37	2-37	2-37#	2-38	2-38	2-38#	2-40	2-40	2-40#
	2-41	2-41	2-41#	2-42	2-42	2-42#	2-43	2-43	2-43#	2-44	2-44	2-44#	2-45	2-45
	2-45#	2-46	2-46	2-46#	2-47	2-47	2-47#	2-48	2-48	2-48#	2-49	2-49	2-49#	2-50
	2-50	2-50#	2-51	2-51	2-51#	2-52	2-52	2-52#	2-53	2-53	2-53#	2-54	2-54	2-54#
	2-55	2-55	2-55#	2-56	2-56	2-56#	2-57	2-57	2-57#	2-58	2-58	2-58#	2-59	2-59
	2-59#	2-60	2-60	2-60#	2-61	2-61	2-61#	2-62	2-62	2-62#	2-63	2-63	2-63#	2-64
	2-64	2-64#	2-66	2-66	2-66#	2-67	2-67	2-67#	2-68	2-68	2-68#	2-69	2-69	2-69#
	2-70	2-70	2-70#	2-71	2-71	2-71#	2-72	2-72	2-72#	2-73	2-73	2-73#	2-74	2-74
	2-74#	2-75	2-75	2-75#	2-76	2-76	2-76#	2-77	2-77	2-77#	2-78	2-78	2-78#	2-79
	2-79	2-79#	2-80	2-80	2-80#	2-81	2-81	2-81#	2-82	2-82	2-82#	2-83	2-83	2-83#
	2-84	2-84	2-84#	2-85	2-85	2-85#	2-86	2-86	2-86#	2-87	2-87	2-87#	2-88	2-88
	2-88#	2-90	2-93	2-93	2-93#	2-94	2-94	2-94#	2-95	2-95	2-95#	2-96	2-96	2-96#
	2-97	2-97	2-97#	2-98	2-98	2-98#	2-99	2-99	2-99#	2-100	2-100	2-100#	2-101	2-101
	2-101#	2-102	2-102	2-102#	2-103	2-103	2-103#	2-104	2-104	2-104#	2-105	2-105	2-105#	2-106
	2-106	2-106#	2-107	2-107	2-107#	2-108	2-108	2-108#	2-109	2-109	2-109#	2-110	2-110	2-110#
	2-111	2-111	2-111#	2-112	2-112	2-112#	2-113	2-113	2-113#	2-114	2-114	2-114#	2-115	2-115
	2-115#	2-116	2-116	2-116#	2-117	2-117	2-117#	2-118	2-118	2-118#	2-119	2-119	2-119#	2-120
	2-120	2-120#	2-121	2-121	2-121#	2-122	2-122	2-122#	2-123	2-123	2-123#	2-124	2-124	2-124#
	2-125	2-125	2-125#	2-126	2-126	2-126#	2-127	2-127	2-127#	2-128	2-128	2-128#	2-129	2-129

CROSS REFERENCE TABLE (CREF V01-03)

	2-129#	2-130	2-130	2-130#	2-131	2-131	2-131#	2-132	2-132	2-132#	2-133	2-133	2-133#	2-134
	2-134	2-134#	2-135	2-135	2-135#	2-136	2-136	2-136#	2-137	2-137	2-137#	2-138	2-138	2-138#
	2-139	2-139	2-139#	2-140	2-140	2-140#	2-141	2-141	2-141#	2-142	2-142	2-142#	2-143	2-143
	2-143#	2-144	2-144	2-144#	2-145	2-145	2-145#	2-146	2-146	2-146#	2-147	2-147	2-147#	2-148
	2-148	2-148#	2-149	2-149	2-149#	2-150	2-150	2-150#	2-151	2-151	2-151#	2-152	2-152	2-152#
	2-153	2-153	2-153#	2-154	2-154	2-154#	2-155	2-155	2-155#	2-156	2-156	2-156#	2-157	2-157
	2-157#	2-158	2-158	2-158#	2-159	2-159	2-159#	2-160	2-160	2-160#	2-161	2-161	2-161#	2-162
	2-162	2-162#	2-163	2-163	2-163#	2-164	2-164	2-164#	2-165	2-165	2-165#	2-166	2-166	2-166#
	2-167	2-167	2-167#	2-168	2-168	2-168#	2-169	2-169	2-169#	2-170	2-170	2-170#	2-171	2-171
	2-171#	2-172	2-172	2-172#	2-173	2-173	2-173#	2-174	2-174	2-174#	2-175	2-175	2-175#	2-176
	2-176	2-176#	2-177	2-177	2-177#	2-178	2-178	2-178#	2-179	2-179	2-179#	2-180	2-180	2-180#
	2-181	2-181	2-181#	2-182	2-182	2-182#	2-183	2-183	2-183#	2-184	2-184	2-184#	2-185	2-185
	2-185#	2-186	2-186	2-186#	2-187	2-187	2-187#	2-188	2-188	2-188#	2-189	2-189	2-189#	2-190
	2-190	2-190#	2-192											
TLICOD	2-130#	2-130#												
TLIST	2-130	2-130#												
TO2COD	2-143#	2-143#												
TOCOD	2-142#	2-142#												
TRACE	2-160	2-160#												
TRACOD	2-160#	2-160#												
TRIG	2-78	2-78#	2-79	2-79#	2-80	2-80#								
TRN	2-28	2-28#												
TRNCOD	2-28#	2-28#												
TYPCOD	2-66#	2-66#												
TYPE	2-66	2-66#												
UNICOD	2-125#	2-125#												
UNIT	2-125	2-125#												
UP	2-18	2-18#												
UPCOD	2-18#	2-18#												
USICOD	2-153#	2-153#												
USING	2-153	2-153#												
VAL	2-33	2-33#												
VALCOD	2-33#	2-33#												
VIECOD	2-116#	2-116#												
VIEW	2-116	2-116#												
WAICOD	2-135#	2-135#												
WAIT	2-135	2-135#												
WBYCOD	2-128#	2-128#												
WBYTE	2-128	2-128#												
WINCOD	2-115#	2-115#												
WINDOW	2-115	2-115#												
WRICOD	2-112#	2-112#												
WRITE	2-112	2-112#												
WRK	2-3	2-3#	2-4	2-4#	2-5	2-5#	2-6	2-6#	2-7	2-7#	2-8	2-8#	2-9	2-9#
	2-10	2-10#	2-11	2-11#	2-12	2-12#	2-13	2-13#	2-14	2-14#	2-15	2-15#	2-16	2-16#
	2-17	2-17#	2-18	2-18#	2-19	2-19#	2-20	2-20#	2-21	2-21#	2-22	2-22#	2-23	2-23#
	2-24	2-24#	2-25	2-25#	2-26	2-26#	2-27	2-27#	2-28	2-28#	2-29	2-29#	2-30	2-30#
	2-31	2-31#	2-32	2-32#	2-33	2-33#	2-34	2-34#	2-35	2-35#	2-36	2-36#	2-37	2-37#
	2-38	2-38#	2-66	2-66#	2-67	2-67#	2-68	2-68#	2-69	2-69#	2-70	2-70#	2-71	2-71#
	2-72	2-72#	2-73	2-73#	2-74	2-74#	2-75	2-75#	2-76	2-76#	2-77	2-77#	2-78	2-78#
	2-79	2-79#	2-80	2-80#	2-81	2-81#	2-82	2-82#	2-83	2-83#	2-84	2-84#	2-85	2-85#
	2-86	2-86#	2-87	2-87#	2-88	2-88#	2-93	2-93#	2-94	2-94#	2-95	2-95#	2-96	2-96#
	2-97	2-97#	2-98	2-98#	2-99	2-99#	2-100	2-100#	2-101	2-101#	2-102	2-102#	2-103	2-103#
	2-104	2-104#	2-105	2-105#	2-106	2-106#	2-107	2-107#	2-108	2-108#	2-109	2-109#	2-110	2-110#
	2-111	2-111#	2-112	2-112#	2-113	2-113#	2-114	2-114#	2-115	2-115#	2-116	2-116#	2-117	2-117#
	2-118	2-118#	2-119	2-119#	2-120	2-120#	2-121	2-121#	2-122	2-122#	2-123	2-123#	2-124	2-124#
	2-125	2-125#	2-126	2-126#	2-127	2-127#	2-128	2-128#	2-129	2-129#	2-130	2-130#	2-131	2-131#

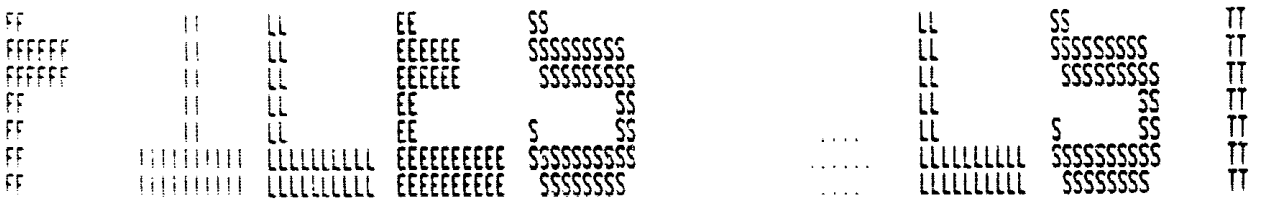
CROSS REFERENCE TABLE (CREF V01-03)

2-132	2-132#	2-133	2-133#	2-134	2-134#	2-135	2-135#	2-136	2-136#	2-137	2-137#	2-138	2-138#
2-139	2-139#	2-140	2-140#	2-141	2-141#	2-142	2-142#	2-143	2-143#	2-144	2-144#	2-145	2-145#
2-146	2-146#	2-147	2-147#	2-148	2-148#	2-149	2-149#	2-150	2-150#	2-151	2-151#	2-152	2-152#
2-153	2-153#	2-154	2-154#	2-155	2-155#	2-156	2-156#	2-157	2-157#	2-158	2-158#	2-159	2-159#
2-160	2-160#	2-161	2-161#	2-162	2-162#	2-163	2-163#	2-164	2-164#	2-165	2-165#	2-166	2-166#
2-167	2-167#	2-168	2-168#	2-169	2-169#	2-170	2-170#	2-171	2-171#	2-172	2-172#	2-173	2-173#
2-174	2-174#	2-175	2-175#	2-176	2-176#	2-177	2-177#	2-178	2-178#	2-179	2-179#	2-180	2-180#
2-181	2-181#	2-182	2-182#	2-183	2-183#	2-184	2-184#	2-185	2-185#	2-186	2-186#	2-187	2-187#
2-188	2-188#	2-189	2-189#	2-190	2-190#								

CROSS REFERENCE TABLE (CREF V01-03)

SEI	1-3#													
2	1-41#	2-3	2-4	2-5	2-6	2-7	2-8	2-9	2-10	2-11	2-12	2-13	2-14	2-15
	2-16	2-17	2-18	2-19	2-20	2-21	2-22	2-23	2-24	2-25	2-26	2-27	2-28	2-29
	2-30	2-31	2-32	2-33	2-34	2-35	2-36	2-37	2-38	2-40	2-41	2-42	2-43	2-44
	2-45	2-46	2-47	2-48	2-49	2-50	2-51	2-52	2-53	2-54	2-55	2-56	2-57	2-58
	2-59	2-60	2-61	2-62	2-63	2-64	2-65	2-67	2-68	2-69	2-70	2-71	2-72	2-73
	2-74	2-75	2-76	2-77	2-78	2-79	2-80	2-81	2-82	2-83	2-84	2-85	2-86	2-87
	2-88	2-93	2-94	2-95	2-96	2-97	2-98	2-99	2-100	2-101	2-102	2-103	2-104	2-105
	2-106	2-107	2-108	2-109	2-110	2-111	2-112	2-113	2-114	2-115	2-116	2-117	2-118	2-119
	2-120	2-121	2-122	2-123	2-124	2-125	2-126	2-127	2-128	2-129	2-130	2-131	2-132	2-133
	2-134	2-135	2-136	2-137	2-138	2-139	2-140	2-141	2-142	2-143	2-144	2-145	2-146	2-147
	2-148	2-149	2-150	2-151	2-152	2-153	2-154	2-155	2-156	2-157	2-158	2-159	2-160	2-161
	2-162	2-163	2-164	2-165	2-166	2-167	2-168	2-169	2-170	2-171	2-172	2-173	2-174	2-175
	2-176	2-177	2-178	2-179	2-180	2-181	2-182	2-183	2-184	2-185	2-186	2-187	2-188	2-189
	2-190													

FFFFFFFFFF IIIIIIIIII LL EEEEEEEEEE SSSSSSSS LL SSSSSSS TTTTTTTTTT
FFFFFFFFFF IIIIIIIIII LL EEEEEEEEEE SSSSSSSSSS LL SSSSSSSSS TTTTTTTTTT
FF II LL EE SS S LL SS S TT


FF II LL EE SS LL SS TT
FFFFFF II LL EEEEEE SSSSSSSS LL SSSSSSSS TT
FFFFFF II LL EEEEEE SSSSSSSS LL SSSSSSSS TT
FF II LL EE SS LL SS TT
FF II LL EE S SS .... LL S SS TT
FF IIIIIIIIII LLLLLLLLLL EEEEEEEEEE SSSSSSSSS ..... LLLLLLLLLL SSSSSSSSS TT
FF IIIIIIIIII LLLLLLLLLL EEEEEEEEEE SSSSSSS .... LLLLLLLLLL SSSSSSS TT

14-OCT-76

FILES THE FILE PROCESSING SYS RT-11 MMAC VM02-10 14-OCT-76 01:32:22
TABLE OF CONTENTS

```
                        .TITLE  FILES  THE FILE PROCESSING SYSTEM
                        .IDENT  /SRM009/
                    ;**********************************************************
                    ;       GLOBAL ITEMS
                    ;
                        .GLOBL  PUTBYT  ; PLACE ONE BYTE IN OUTPUT BUFFER
                        .GLOBL  SNDBFR  ; OUTPUT BUFFER WITH EOI
                        .GLOBL  ADRDEV  ; ADDRESS ROUTINE
                        .GLOBL  BFRALC  ; BUFFER ALOCATOR
                        .GLOBL  UNADR   ; CLEAN UP DEVICE
                        .GLOBL  PRIVAL  ; FORMS ASCII NUMBER IN BUFFER
                        .GLOBL  PRISTG  ; FORMS ASCII STRING IN BUFFER
                        .GLOBL  DIMSTP  ; DIMENSION A STRING
                        .GLOBL  LOCTG   ; LOCATES A STACK TAG
                        .GLOBL  INPSTG  ; INPUT STRING INTO BUFFER
                        .GLOBL  STKBLD  ; BUILDS AN INVERTED STACK WITH POINTERS
                        .GLOBL  TYPARG  ; FIND TYPE OF ARG ON STACK
                        .GLOBL  FILES   ; FILE PROCESSOR
                        .GLOBL  BACKUP  ; BACKUP STACK ONE ITEM
                        .GLOBL  UPRADR  ; STORAGE OF I/O OPERATION WORD
                        .GLOBL  INT1    ; DIMENSION NUMBER
                        .GLOBL  A5X     ; ADD 5 TO INDEX REG.
```

```
                                        .SBTTL  STKBLD--THE BACKWARD STACK BUILDER
                                ;THIS ROUTINE BUILDS A NEW STACK IN PROPER ORDER
                                ;FOR "NORMAL FIX" NOTATION PROCESSING.  IT USES TYPARG TO
                                ;GET THE NEXT TAG.  IF THAT TAG IS AN EOL THE ROUTINE EXITS.
                                ;IF THE TAG IS SEMITG THE SEMITG IS PUSHED ON THIS NEW STACK
                                ;FOR ROUTINES TO USE LATER.  ALL OTHER ITEMS ARE TAGGED
                                ;AS BAKSTG AND INCLUDES THE RESULTING BYTE FROM TYPARG AND AN
                                ;INTER-STACK POINTER THAT POINTES TO THE ORIGINAL ENTRY
                                ;IT IS CALLED WITH R0 POINTING TO THE ORIGINAL TOP OF STACK
                                ;IT WILL ALSO SKIP ITM2TG (WHICH IS GENERALY ONE OF MY EOL TAGS).
                                ;
                                        .GLOBL  CLRARG
                0030                    AVLD    = H30           ; A MASK FOR TYPARG RETURN INFORMATION
                                ;
0000    32              STKBLD: PUL A                           ; SAVE RETURN ADDRESS
0001    97      01G             STA A   DREXTB+1,D
0003    32                      PUL A
0004    97      02G             STA A   DREXTB+2,D
0006    DE      00G     STKBL.: LDX     R0,D                    ; GET STACK POINTER
0008    A6      01      STKBL:  LDA A   1,X                     ; GET TAG
000A    81      00G             CMP A   EOLTG,I                 ; EOL?
000C    26      03              BNE     STKIT1
000E    7E      0000G   BYEBYE: JMP     DREXTB                  ; EXIT
0011    81      00G     STKIT1: CMP A   ATSNTG,I                ;  LOOKS LIKE EOL
0013    27      F9              BEQ     BYEBYE
0015    81      00G             CMP A   ITM2TG,I                ; MY EOL TAG?
0017    26      0H              BNE     STKSEM
0019    08                      INX
001A    08                      INX
001B    20      C5              BRA     STKLUP
001D    81      00G     STKSEM: CMP A   SEMITG,I                ; IS IT THE SEMICOLON?
001F    26      06              BNE     REALTG
0021    36                      PSH A                           ; BUT SEMICOLON ON MY STACK
0022    08              STKLOP: INX                             ; UPDATE STACK POINTER
0023    DF      00G             STX     R0,D
0025    20      E1              BRA     STKBL                   ; GO FOR MORE ON STACK
0027    96      01G     REALTG: LDA A   R0+1,D                  ; PUSH INTER STACK POINTER
0029    36                      PSH A
002A    96      00G             LDA A   R0,D
002C    36                      PSH A
002D    BD      0000G           JSR     CLRARG                  ; FING OUT WHAT IT IS
0030    36                      PSH A
0031    C6      00G             LDA B   BAKSTG,I                ; A GOOD OLD TAG
0033    37                      PSH B                           ; SAVE IT AS A TAG
0034    81      30              CMP A   AFAIL,I
0036    27      D6              BEQ     BYEBYE                  ; UNDEFINDED ERROR
0038    20      CC              BRA     STKBL                   ; TRY FOR MORE
                                ;XXXXXXXXXTHIS IS THE END FOR NOW
                                ;THERE IS MORE CODE ON THIS FILE IF YOU ERASE THIS "END" STATEMENT
                                ;
```

```
                                        .SBTTL  FILES---FILE SYSTEM DISPATCH PROCESSOR
                                        .GLOBL  FILEIN,FILEOT,FILES,FIL,TYPFIL,FILERQ
                                        .GLOBL  DISKCL          ; CALL FILE SYS. ROM
                                ;
                                ;REQUEST FOR I/O
                                ;LOOK AT A.STAT TO DETERMINE DIRECTION
                                ;
003A                            FILEIN:
003A    86      06              FILEOT: LDA A   6.,I
003C    20      0A                      BRA     FILERQ
                                ;
                                ;COMMAND REQUEST
                                ;LOOK AT OPRADR TO DETERMINE COMMAND
                                ;
003E    86      03              FILES:  LDA A   3.,I
0040    20      06                      BRA     FILERQ
                                ;
                                ;# REQUEST
                                ;INFORMATION IS ON STACK
                                ;
0042    86      09              FIL:    LDA A   9.,I
0044    20      02                      BRA     FILERQ
                                ;
                                ;SPECIAL ENTRY POINT FOR OLD, SAVE, LIST
                                ;DEFINED IN MTCTL AS LDA A 12.,I
                                ;
                                ;--------
                                ;
                                ;TYPE FUNCTION REQUEST
                                ;
0046    86      0F              TYPFIL: LDA A   15.,I
0048    7E      0000G           FILERQ: JMP     DISKCL
                0001'                   .END
```

```
FILES  THE FILE PROCESSING SYS RT-11 MACR  VM02-10   14-OCT-76 01:32:22 PAGE 3+
SYMBOL TABLE

ABRFLG=  0040        ADRDEV= XXXXXX G     AFAIL =  0030        AMAT  =  0010        ARRAY =  0020
ASTR  =  0020        ATLOD =  0008        ATSNTG= XXXXXX G     ATWLD =  0004        AWLD  =  0030
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     BACKUP= XXXXXX G     BAKSTG= XXXXXX G
BANK  = XXXXXX G     BERRALC= XXXXXX G    BERSIT=  0020        BLINK = XXXXXX G     BMAT  =  0004
BRKCNT= XXXXXX G     BSTR  =  0008        BUSACT=  0010        BYEBYE   000ER       CDOPTR= XXXXXX G
CDSPTR= XXXXXX G     CHAR  = XXXXXX G     CHRCNT= XXXXXX G     CLPTR = XXXXXX G     CLRARG= XXXXXX G
CMAT  =  0001        COLCNT= XXXXXX G     CRDC3 =  0002        CREOF =  0008        CREOL =  0004
CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT= XXXXXX G     CRVLD =  0080
CSTR  =  0002        CTKN  = XXXXXX G     CURSOR= XXXXXX G     DATDEV=  0022        DIMFLG=  0004
DIMSTR= XXXXXX G     DIRCT =  0080        DISCNT= XXXXXX G     DISKCL= XXXXXX G     DISSRC=  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     EREBER= XXXXXX G
ERFILE= XXXXXX G     ERIOE = XXXXXX G     ERNIOD= XXXXXX G     ERNSEP= XXXXXX G     ERPRCD = XXXXXX G
ERTERM= XXXXXX G     ERUNDF= XXXXXX G     ESTG  = XXXXXX G     EXTFLG=  0080        FIL      0042RG
FILDEV=  0000        FILEIN   003BRG      FILEOT   003BRG      FILERO   0048RG      FILES    003ERG
FIX1A =  0003        FIX1B =  0004        FMTWLD=  0008        FNFLG =  0010        FORTG = XXXXXX G
GLBFLG= XXXXXX G     GOSTG = XXXXXX G     IMXTG = XXXXXX G     INPSTG= XXXXXX G     INPUTK=  0001
INT   = XXXXXX G     IOBER1= XXXXXX G     IOELGS= 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     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
MTPD2 =  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=  0005        NULLTG= XXXXXX G     OBJATR=  0002
OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        ONSFLG=  0002        OPRADR= XXXXXX G
PAETG = XXXXXX G     PARM  =  0008        PGMATR=  0002        PGMBP =  0005        PGMCD =  0009
PGMFP =  0003        PGMLEN=  0000        PGMLNN=  0007        PGMPTR= XXXXXX G     PGMTG = XXXXXX G
PLOSTG= XXXXXX G     PNDEOE= XXXXXX G     PNDFLG= XXXXXX G     PNTNTG= XXXXXX G     PNTSTG= XXXXXX G
POINT = XXXXXX G     PPMODE= XXXXXX G     PRIDEF=  0001        PRISTG= XXXXXX G     PRIVAL= XXXXXX G
PRTTG = XXXXXX G     PSCTG = XXXXXX G     PUTBYT= XXXXXX G     REALTG   0027R       RECLFG=  0004
RFAIL =  00C0        RMAT  =  0040        RPTCTL=  008G        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        SEMITG= XXXXXX G
SNOBFR= XXXXXX G     SNOIT =  0080        STAT37= XXXXXX G     STKBL    0006R       STKBLD   0000RG
STKBL    0006R       STKIT1   0011R       STKLOP   0022R       STKSEM   0010R       STPFLG=  0040
STPKEY=  0020        STRING=  0010        SYSERR= XXXXXX G     TABPTR= XXXXXX G     TABPTR= XXXXXX G
TCOL  = XXXXXX G     TRCFLG=  0020        TYPARG= XXXXXX G     TYPFIL   0046RG      TSTDEV=  0025
UNADR = XXXXXX G     UNDEF =  0080        VALTG = XXXXXX G     VALUND=  0040        XAXIS = XXXXXX G

. ABS.   0000      00
         0048      01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE  2575  WORDS
.SY:FILES/C(DK1:SETCLI.FILES
```

CROSS REFERENCE TABLE (CREF VO1.07)

CROSS REFERENCE TABLE (CREF MC, ...)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-3#

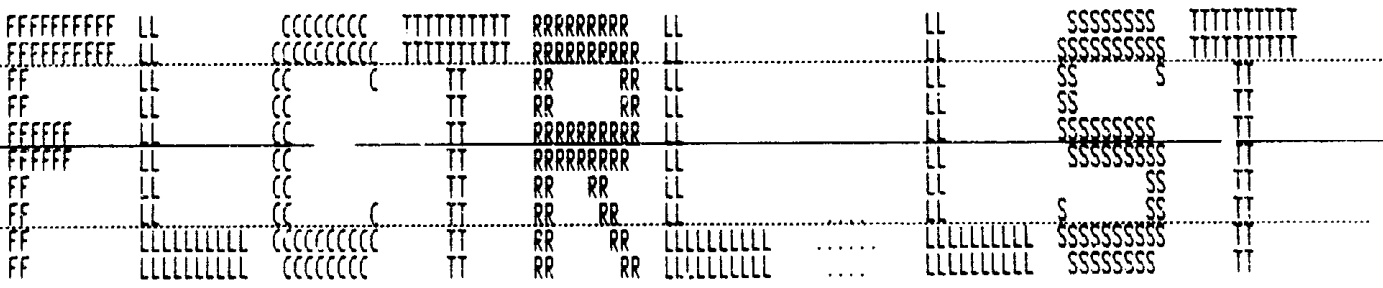


TABLE OF CONTENTS

```
                                        .TITLE  FLCTRL  FLOW OF CONTROL COMMANDS
                                        .IDENT  /BE3032/
                                ;
                                        .GLOBL  FLCTRL
0000'                           FLCTRL: ;
                                        .GLOBL  GETLN,PUSHES,POPES,LOCTG,LOCTGR,BACKUP
                                        .GLOBL  TYPARG,SETARG,CLRARG
                                        .GLOBL  RESTZ,IDLE,END,CLOSE
                                        .GLOBL  FIX1,TRUTH,T4,FPADD,FPSUB,FUZZIE,ZX
                                        .GLOBL  PSHFPN,PULFPN,FFONE
                                        .GLOBL  A5X,A6X,A7X,A8X,A9X,A11X,D8X,D9X
                                        .GLOBL  SETERR,SETERA,HALTA
                                ;
                                        .SBTTL  XXX RUN         RUN COMMAND
                                        .GLOBL  RUN
                                ;
                                ;       INPUTS
                                ;               OPTIONAL LINE NO. ON STACK
                                ;
                                ;       OUTPUTS
                                ;               RUN MODE IS SET
                                ;
                                ;       NOTES
                                ;               IF NO LINE NUMBER IS GIVEN THE STACK IS RESET.
                                ;
0000  30                        RUN:    TSX                     ;STATEMENT NO. GIVEN
0001  A6  00                           LDA A   0,X
0003  81  00G                           CMP A   VALTG,I
0005  26  1B                            BNE     RUNNLN          ;NO LINE NO. GIVEN
0007  BD  0000G                         JSR     GETLN           ;CONVERT VALUE TO LINE PTR
000A  96  0FG                           LDA A   ERRCD,D         ;ERROR CODE SET
000C  26  11                            BNE     RUNEXT
000E  DE  00G                           LDX     R0,D            ;SET UP NEXT LN PTR
0010  4F                                CLR A                   ;NO CHANGE IN KEY STATUS
0011  DF  00G                   RUNA:   STX     NLPTR,D
0013  9A  00G                           ORA A   GLBFLG,D
0015  97  00G                           STA A   GLBFLG,D
0017  85  40                            BIT A   STPFLG,I        ;DON'T RUN IN STEP MODE
0019  26  04                            BNE     RUNEXT
001B  8A  80                            ORA A   RUNFLG,I        ;SET RUN MODE
001D  97  00G                           STA A   GLBFLG,D
001F  7E  0000G                 RUNEXT: JMP     DREXTA
                                ;
0022  9E  00G                   RUNNLN: LDS     SBP,D           ;GET END OF STACK
0024  86  00G                           LDA A   EOSTG,I         ;MARK AS END OF STACK
0026  36                                PSH A
0027  BD  0000G                         JSR     RESTZ           ;RESET DATA STMT PTRS
002A  DE  00G                           LDX     PGMPTR,D        ;START AT ORIGIN OF PROGRAM
002C  86  10                            LDA A   KEYFLG,I        ;SET RUN AND DISABLE KEYS
002E  8D  03                            BSR     RUNB
0030  7E  0000G                         JMP     IDLE            ;IF I EVER GET HERE STOP THE SYSTEM
                                ;
0033  C6  00G                   RUNB:   LDA B   EOLTG,I         ;FAKE END OF LINE
0035  37                                PSH B
0036  20  D9                            BRA     RUNA            ;JOIN COMMON CODE
```

```
                                        .SBTTL  XXX STOP        STOP COMMAND
                                        .GLOBL  STOP
                                ;
                                ;       ALL STOP DOES IS TO SET THE ERROR CODE VARIABLE TO ERSTOP
                                ;       THIS IS THE EASIEST WAY TO STOP THE EVALUATOR
                                ;
0038    BD      0000G           STOP:   JSR     SETERR          ;SET ERROR CODE FOR STOP
0038    00G                             .BYTE   ERSTOP
```

```
                                        .SBTTL  *** GOTO         GO TO STATEMENT
                                        .GLOBL  GOTO
                                ;
                                ;       INPUTS
                                ;               STATEMENT NUMBER ON THE STACK
                                ;
                                ;       OUTPUTS
                                ;               NLPTR IS ALTERED
                                ;
003C  BD  0000G   GOTO:  JSR    GETLN           ;CONVERT LN NO TO LN PTR
003F  96  00G            LDA A  ERRCD.D         ;DID IT WORK
0041  26  04             BNE    GOTEXT          ;NO
0043  DE  00G            LDX    R0.D            ;FIX UP NEXT LN PTR
0045  DF  00G            STX    NLPTR.D
0047  7E  0000G  GOTEXT: JMP    DREXTA          ;GO TO CALLER
```

```
                                        .SBTTL  XXX GOSUB       GOSUB STATEMENT
                                        .GLOBL  GOSUB
                                ;
                                ;       INPUTS
                                ;               LINE NUMBER ON STACK
                                ;
                                ;       OUTPUTS
                                ;               NLPTR IS ALTERED
                                ;               RETURN ENTRY IS STACKED
                                ;
004A    80      0000G   GOSUB:  JSR     GETLN           ;CONVERT LN NO TO LN PTR
004D    96      ??      GOSUB:  LDA A   ERRCD,D         ;DID IT WORK
004F    26      10              BNE     GOSEXT          ;NO IT DID NOT
0051    DE      00G             LDX     R0,D            ;FIX UP NEXT LN PTR
0053    DF      00G             STX     NLPTR,D
0055    96      01G             LDA A   CLPTR+1,D       ;PUT ON LINE PTR
0057    D6      00G             LDA B   CLPTR,D
0059    DE      00G             LDX     CLPTR,D         ;SEE IF I IN IMEX MODE
005B    EE      07              LDX     PGMLNN,X        ;LINE NO. ZERO
005D    26      02              BNE     GOSSKP
005F    4F                      CLR A
0060    5F                      CLR B
0061    36              GOSSKP: PSH A
0062    37                      PSH B
0063    86      00G             LDA A   GOSTG,I
0065    36                      PSH A
0066    80      03      FIXUP:  BSR     GOSTAG          ;FIX UP EOL ON STACK
0068    7E      0000G           JMP     IDLE            ;IF I GET HERE GO TO IDLE LOOP
006B    86      00G     GOSTAG: LDA A   EOLTG,I         ;NOW TAG ENTRY
006D    36                      PSH A
006E    7E      0000G   GOSEXT: JMP     DREXTA          ;RETURN TO CALLER
```

```
                                        .SBTTL  XXX OF          OF TOKEN FOR SELECTION IN GOTO/GOSUB
                                        .GLOBL  OF
                                ;
                                ;       INPUTS
                                ;               LIST OF LINE NUMBERS AND EXP ON STACK
                                ;
                                ;       OUTPUTS
                                ;               LINE NUMBER ON STACK
                                ;
                                ;       NOTES
                                ;               IF NO LINE IS TO BE SELECTED THE LINE IS ABORTED
                                ;
0071  9F    00G       OF:     STS     R0,D            ;WORKING SP INTO R0 (REALY SP-1)
0073  86    00G               LDA A   EOLTG,I         ;FIND END OF STACK
0075  BD    0000G             JSR     LOCTG
0078  DE    00G               LDX     R0,D            ;BACK OFF TO GET EXP VALUE
007A  BD    0000G             JSR     D9X             ;SET UP TYPARG OUTPUT ARE
007D  DF    00G               STX     R0,D
007F  BD    0000G             JSR     CLRARG          ;TEST FOR NUMBER
0082  96    00G               LDA A   R4,D            ;R4=0 FOR VALUE
0084  26    44                BNE     OFERR           ;IF IT FAILED I CAN'T RUN
0086  DE    00G               LDX     R0,D            ;FIX EXP VALUE
0088  BD    0000G             JSR     D8X
008B  BD    0000G             JSR     FIX1
008E  26    34                BNE     OFRST           ;IF IT FAILED JUST FALL OUT OF GOTO/GOSUB
0090  A6    03                LDA A   3,X             ;NUMBER CAN'T BE BIGGER THAN 31
0092  26    33                BNE     OFABR
0094  E6    04                LDA B   4,X             ;ZERO IS ALSO INVAL'D
0096  27    2F                BEQ     OFABR
0098  43                      COM A                   ;STACKS ARE BACKWARDS
0099  53                      COM B
009A  6C    04                INC     4,X             ;BIAS FOR BACKWARDS ADDRESSING
009C  58                      ASL B                   ;N TIMES 9 = N TIMES 8 PLUS N
009D  58                      ASL B
009E  58                      ASL B
009F  49                      ROL A
00A0  E0    04                SUB B   4,X             ;ADD N
00A2  82    00                SBC A   0,I
00A4  DB    01G               ADD B   R0+1,D          ;NOW FIND ADDR OF LINE NO.
00A6  99    00G               ADC A   R0,D
00A8  9F    00G               STS     R1,D            ;IF BELOW SP FORGET IT
00AA  97    00G               STA A   R2,D            ;MOVE AB TO SP
00AC  D7    01G               STA B   R2+1,D
00AE  D0    01G               SUB B   R1+1,D          ;SEE IF LESS THEN SP
00B0  92    00G               SBC A   R1,D
00B2  2B    13                BMI     OFABR
00B4  9E    00G               LDS     R2,D            ;SP NOW POINTS TO BRANCH LINE NUMBER
00B6  CE    0000G             LDX     R4,I            ;SAVE LINE NUMBER
00B9  BD    0000G             JSR     PULFPN
00BC  9E    00G               LDS     R0,D            ;CLEAN UP STACK
00BE  BD    0000G             JSR     PSHFPN
00C1  7E    0000G     OFEXIT: JMP     DREXTA
                                ;
00C4  7F    0000G     OFRST:  CLR     ERRCD           ;RESET ERROR CODE IF FAILED IN FIX1
00C7  7E    0000G     OFABR:  JMP     HALTA
                                ;
00CA  BD    0000G     OFERR:  JSR     SETERA
```

XXX OF OF TOKEN FOR SELECTION IN GOTO/GOSUB

```
    58          COCD      OOG                         .BYTE   EROFA
```

```
                                .SBTTL  XXX RETURN      BASIC RETURN STATEMENT
                                .GLOBL  RETURN
                                ;
                                ;       INPUTS
                                ;               RETURN POINTS ON STACK
                                ;
                                ;       OUTPUTS
                                ;               EVAL CONTROLS ARE SET UP TO DO RETURN
                                ;
                                ;       NOTES
                                ;               IF NO RETURN IS ON STACK CONTROL GOES TO MONITOR
                                ;
00CE 9F  00G    RETURN: STS     R0,D            ;STARTING LOC FOR LOCTGR
0000 86  00G            LDA A   ESTG,I          ;EVAL STATUS IS MIN VALUE
0002 C6  00G            LDA B   GOSTG,I         ;GOSUB/RETURN IS MAX
0004 BD  0000G          JSR     LOCTGR
0007 DE  00G            LDX     R0,D            ;IF=0 EOS WAS FOUND
0009 27  1F             BEQ     RTNIDL
0008 A6  01             LDA A   1,X             ;GET TAG THAT WAS FOUND
000D 81  00G            CMP A   GOSTG,I         ;IS IT GOSUB/RETURN
000F 27  05             BEQ     RTNGOS
00E1 DE  00G            LDX     R0,D            ;CLEAN UP STACK
00E3 09                 DEX                     ;GO BACK TO EOL
00E4 35                 TXS
00E5 39                 RTS
                        ;
00E6 9E  00G    RTNGOS: LDS     R0,D            ;SET SP TO GOSUB/RETURN ENTRY
00E8 32                 PUL A                   ;SKIP TAG
00E9 32                 PUL A                   ;POP GOSUB LINE PTR
00EA 97  00G            STA A   CLPTR,D         ;EVAL WILL NOT EXEC CURRENT LINE
00EC 32                 PUL A
00ED 97  01G            STA A   CLPTR+1,D
00EF DE  00G            LDX     CLPTR,D         ;LINK TO NEW LINE
00F1 27  02             BEQ     RTNEXT          ;IF NULL FORGET CHAIN
00F3 EE  03             LDX     PGMEP_X         ;NEXT LINE ADDRESS
00F5 DF  00G    RTNEXT: STX     NLPTR,D         ;  FROM CLPTR
00F7 7E  0000G          JMP     HALTA           ;ABORT LINE AND EXIT-A
                        ;
00FA 7E  0000G  RTNIDL: JMP     END             ;DIRTY EXIT TO IDLE LOOP & MONITOR
```

```
                                        .SBTTL  XXX IF          BASIC IF STATEMENT
                                        .GLOBL  IF
                                ;
                                ;       INPUTS
                                ;               LINE NO AND EXP ON STACK
                                ;
                                ;       OUTPUTS
                                ;               IF EXP IS TRUE EVAL CONTROLS ARE SET
                                ;                 UP TO BRANCH TO GIVEN LINE
                                ;
00FD    BD      0000G   IF:     JSR     GETLN           ;SEE IF LINE NO. IS VALID
0100    96      00G             LDA A   ERRCD,D
0102    26      09              BNE     IFABR           ;CONVERSION ERROR
0104    BD      0000G           JSR     TRUTH           ;TEST FOR TRUE
0107    27      04              BEQ     IFABk           ;IF=0 ABORT LINE
0109    DE      00G             LDX     RO,D            ;ALTER NEXT LINE PTR
0108    DF      00G             STX     NLPTR,D
010D    7E      0000G   IFABR:  JMP     HALTA
```

```
                                        .SBTTL  XXX FOR         FOR STATEMENT
                                        .GLOBL  FOR
                                ;
                                ;       INPUTS
                                ;               STOP VALUE ON STACK
                                ;               INCREMENT ON STACK (OPTIONAL)
                                ;               CLPTR FROM EVAL CONTROLS
                                ;
                                ;       OUTPUTS
                                ;               STACK ENTRY FOR FOR/NEXT LOOP
                                ;
0110  96  00G       FOR:      LDA A   GLBFLG,D        ;SEE IF I AM IN IMMEDIATE EXEC MODE
0112  85  C0                  BIT A   192.,I
0114  26  04                  BNE     FOROK
0116  8D  0000G               JSR     SETERA
0119  00G                     .BYTE   ERNIMX
                                ;
011A  DE  00G       FOROK:    LDX     CLPTR,D         ;TEST FOR SIMPLE VARIABLE
011C  EE  0A                  LDX     PGMCD+1,X       ;HAVE ADDR OF NT ENTRY NOW
011E  A6  04                  LDA A   NTATTR,X
0120  85  40                  BIT A   SCALER,I        ;IT MUST BE A SCALER
0122  27  28                  BEQ     FORERR          ;IT IS DEFINED BY ASSIGN IN EXP EVAL
                                                      ;SET UP TO TEST INFO STILL ON STACK
0124  8D  0000G               JSR     SETARG
0127  8D  0000G               JSR     TYPARG
0129  96  00G                 LDA A   R4,D            ;IF ZERO THERE WERE 2 VALUES GIVEN
012C  27  0A                  BEQ     FORPV
012E  81  30                  CMP A   AFAIL,I         ;STEP MUST BE ERROR, NOT MATRIX
0130  26  1A                  BNE     FORERR
0132  CE  0000G               LDX     FPONE,I         ;PUSH A ONE ON THESTACK
0135  8D  0000G               JSR     PSHFPN
0138  DE  00G       FORPV:    LDX     CLPTR,D         ;PUSH PTR TO VARIABLE
013A  A6  08                  LDA A   PGMCD+2,X
013C  36                      PSH A
013D  A6  0A                  LDA A   PGMCD+1,X
013F  36                      PSH A
0140  96  01G                 LDA A   CLPTR+1,D       ;PUSH FOR LINE PTR
0142  36                      PSH A
0143  96  00G                 LDA A   CLPTR,D
0145  36                      PSH A
0146  86  00G                 LDA A   FORTG,I         ;TAG STACK ENTRY
0148  36                      PSH A
0149  7E  0066'               JMP     FIXUP
                                ;
014C  8D  0000G     FORERR:   JSR     SETERA
014F  00G                     .BYTE   ERFORA
```

```
                                   .SBTTL  *** NEXT        NEXT STATEMENT
                                   .GLOBL  NEXT
                          ;
                          ;        INPUTS
                          ;                VARIABLE PTR ON STACK
                          ;                FOR/NEXT ENTRY ON STACK
                          ;
                          ;        OUTPUTS
                          ;                UPDATED VARIABLE AND BRANCH OR CLEAN STACK
                          ;
            0000          NXTAG   =        0.           ;DISPLACE TO TAG
            0001          NXFOR   =        NXTAG+1.     ;PTR TO FOR LINE
            0003          NXNT    =        NXFOR+2.     ;NAME TABLE PTR FOR VARIABLE
            0005          NXINCR  =        NXNT+2.      ;INCREMENT
            000E          NXSTOP  =        NXINCR+9.    ;LAST VALUE
                          ;
0150   9F   00G           NEXT:   STS      R0.D         ;SET UP FOR LOCTG
0152   30                         TSX                   ;GET PTR TO N. T. ENTRY
0153   EE   01                    LDX      1,X
0155   DF   00G                   STX      R7.D         ;SAVE NT PTR AS VAR ID
0157   A6   04                    LDA A    NTATTR,X     ;TEST FOR SCALER
0159   88   40                    EOR A    SCALER,I
015B   85   C0                    BIT A    SCALER+UNDEF,I          ;DID IT GET ERASED
015D   27   07                    BEQ      NXTOK
015F   BD   0000G         NXTERR: JSR      SETERA
0162   00G                        .BYTE    ERNXTA
                          ;
0163   BD   0000G         NXTLP:  JSR      BACKUP       ;SKIP THIS FOR ENTRY
0166   86   00G           NXTOK:  LDA A    FORT,I       ;TAG I WANT
0168   BD   0000G                 JSR      LOCTG
016B   DE   00G                   LDX      R0.D         ;IF ZERO I HIT END OF STACK
016D   27   E0                    BEQ      NXTERR
016F   EE   04                    LDX      NXNT+1,X     ;GET N. T. PTR FOR VAR
0171   9C   00G                   CPX      R7.D         ;IF PTRS AGREE IT IS THE SAME VAR
0173   26   EE                    BNE      NXTLP
0175   BD   0000G                 JSR      A5X          ;PSH CRNT VALUE - X IS NT PTR
0178   BD   0000G                 JSR      PSHFPN
017B   DE   00G                   LDX      R0.D         ;PUSH INCR FROM FOR STACK ENTRY
017D   BD   0000G                 JSR      A7X
0180   BD   0000G                 JSR      PSHFPN
0183   BD   0000G                 JSR      FPADD        ;ADD INCR TO CRNT VALUE
0186   DE   00G                   LDX      R7.D         ;PUT RESULT INTO N. T.
0188   BD   0000G                 JSR      A5X
018B   BD   0000G                 JSR      PULFPN
018E   BD   0000G                 JSR      PSHFPN       ;STILL NEED COPY
0191   DE   00G                   LDX      R0.D         ;PUSH STOP VALUE
0193   BD   0000G                 JSR      A7X
0196   BD   0000G                 JSR      A9X
0199   BD   0000G                 JSR      PSHFPN
019C   BD   0000G                 JSR      FUZZIE       ;TEST ANSWER
019F   DE   00G                   LDX      R0.D
01A1   A6   07                    LDA A    NXINCR+2,X   ;TEST LOOP TO SEE IF I AM GOING BACKWARDS
01A3   2B   06                    BMI      NXTBCK
01A5   96   00G                   LDA A    T4.D         ;CRNT GE STOP
01A7   2E   18                    BGT      NXTEND
01A9   20   04                    BRA      NXTSKP
                          ;
```

```
    58  01A8  96  00G       NXTBCK: LDA A  T4,D
    59  01AD  28  12                BMI    NXTEND
    60                               ;
    61  01AF  DE  00G       NXTSKP: LDX    R0,D
    62  01B1  09                    DEX
    63  01B2  09                    DEX
    64  01B3  35                    TXS                    ;CLEAN UP STACK
    65  01B4  EE  04                LDX    NXFOR+3,X       ;GET FOR LINE PTR
    66  01B6  DF  00G               STX    CLPTR,D
    67  01B8  27  02                BEQ    NXTEXT          ;NO MORE LINES
    68  01BA  EE  03                LDX    PGMFP,X
    69  01BC  DF  00G       NXTEXT: STX    NLPTR,D
    70  01BE  7E  0000G     NXTABR: JMP    HALTA
    71                               ;
    72  01C1  8D  0000G     NXTEND: JSR    BACKUP          ;PRUNE FOR/NEXT ENTRY FROM STACK
    73  01C4  9E  00G               LDS    R0,D
    74  01C6  20  F6                BRA    NXTABR          ;GO TO LINE AFTER NEXT THIS TIME
    75                               ;
    76          0001'                    .END
```

```
FLCTRL FLOW OF CONTROL COMMAND RT-11 MMAC  VM02-10   14-OCT-76 01:32:31 PAGE 15+
SYMBOL TABLE

ABRFLG=  0040        AFAIL =  0030        ALLOK =  0004        ALLTG = XXXXXX G     AMAT  =  0010
ARRAY =  0020        ASGCOD= XXXXXX G     ASTR  =  0020        ATSNTG= XXXXXX G     A11X  = XXXXXX G
A5X   = XXXXXX G     A6X   = XXXXXX G     A7X   = XXXXXX G     A8X   = XXXXXX G     A9X   = XXXXXX G
BACKUP= XXXXXX G     BAKSTG= XXXXXX G     BMAT  =  0004        BSTR  =  0008        CALLTG= XXXXXX G
CDOPTR= XXXXXX G     CDSPTR= XXXXXX G     CLOSE = XXXXXX G     CLPTR = XXXXXX G     CLRARG= XXXXXX G
CMAT  =  0001        CONCOD= XXXXXX G     CRCOD = XXXXXX G     CSTR  =  0002        CTKN  = XXXXXX G
DATCOD= XXXXXX G     DIMFLG=  0004        DISSRQ=  0008        DREXTA= XXXXXX G     DREXTB= XXXXXX G
D8X   = XXXXXX G     D9X   = XXXXXX G     E   = XXXXXX G       EOFCOD= XXXXXX G     EOFTBL= XXXXXX G
EOLTG = XXXXXX G     EOSTG = XXXXXX G     E  OD= XXXXXX G      ERASGN= XXXXXX G     ERBRK = XXXXXX G
ERDOMN= XXXXXX G     EREOFN= XXXXXX G     E  ORA= XXXXXX G     ERLNNF= XXXXXX G     ERNDT = XXXXXX G
ERNIMX= XXXXXX G     ERNOEN= XXXXXX G     ERNXFN= XXXXXX G     ERNXTA= XXXXXX G     EROFA = XXXXXX G
ERRCD = XXXXXX G     ERRCD8= XXXXXX G     ERSHAP= XXXXXX G     ERSTOP= XXXXXX G     ERUNDF= XXXXX. G
ERVAL = XXXXXX G     ERWSFL= XXXXXX G     ESTG  = XXXXXX G     EXTFLG=  0080        FIXUP    0066R
FIX1  = XXXXXX G     FLCTRL=  0000RG      FNACOD= XXXXXX G     FNFLG =  0010        FNTBL = XXXXXX G
FOR      0110RG      FORERR   014CR       FOROK    011AR       FORPV    0138R       FORTG = XXXXXX G
FPADD = XXXXXX G     FPONE = XXXXXX G     FPSUB = XXXXXX G     FUZZIE= XXXXXX G     GETLN = XXXXXX G
GLBFLG= XXXXXX G     GOSEXT   006ER       GOSSKP   0061R       GOSTAG   0068R       GOSTG = XXXXXX G
GOSUB    004ARG      GOTEXT   0047R       GOTO     003CRG      HALTA = XXXXXX G     IDLE  = XXXXXX G
IF       00FDRG      IFABR    010DR       IMACOD= XXXXXX G     IMXFLG=  0020        IMXTG = XXXXXX G
ITM1TG= XXXXXX G     ITM2TG= XXXXXX G     JMPAX = XXXXXX G     JMPX  = XXXXXX G     KEYFLG=  0010
KEYSTK= XXXXXX G     LBRKTG= XXXXXX G     LCLFLG= XXXXXX G     LDAX  = XXXXXX G     LDBX  = XXXXXX G
LDXX  = XXXXXX G     LISTTG= XXXXXX G     LITCOD= XXXXXX G     LNNOTG= XXXXXX G     LOCTG = XXXXXX G
LOCTGR= XXXXXX G     LSP   = XXXXXX G     LSTCOD= XXXXXX G     MMICOD= XXXXXX G     MPLCOD= XXXXXX G
MULCOD= XXXXXX G     NCSFLG=  0001        NEXT     0150RG      NLPTR = XXXXXX G     NTAPTR=  00J8
NTATTR=  0004        NTDIMS=  0009        NTDLEN=  0005        NTLINK=  0000        NTNAME=  00J2
NTPTR = XXXXXX G     NTSPTR=  000P        NTVAL =  0005        NTWCOL=  0007        NTWLEN=  0007
NTWROW=  0005        NULLTG= XXXXXX G     NXFOR =  0001        NXINCR=  0005        NXNT  =  0003
NXSTOP=  000E        NXTABR   01BER       NXTAG =  0000        NXTBCK   01A8R       NXTEND   01C1R
NXTERR   015FR       NXTEXT   01BCR       NXTLP    0163R       NXTOK    0166R       NXTSKP   01AFR
OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005        OBJLEN=  0000        OF       0071RG
OFABR    00C7R       OFERR    00CAR       OFEXIT   00C1R       OFRST    00C4R       ONSFLG=  0002
ONTBL = XXXXXX G     OPRADR= XXXXXX G     OUAFLG=  0002        PAETG = XXXXXX G     PARM  =  0008
PGMATR=  0002        PGMBP =  0005        PGMCD =  0009        PGMFP =  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     POPES = XXXXXX G     PRTTG = XXXXXX G
PSCTG = XXXXXX G     PSHFPN= XXXXXX G     PULFPN= XXXXXX G     PUSHES= XXXXXX G     RESTZ = XXXXXX G
RETURN   00CERG      RFAIL =  00C0        RMAT  =  0040        RSTR  =  0080        RTNEXT   00F5R
RTNGOS   00E6R       RTNJUL   00FAR       RTRNTG= XXXXXX G     RUN      0000RG      RUNA     0011R
RUNB     0033P       RUNEXT   001FR       RUNFLG=  0080        RUNNLN   0022R       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     SBP   = XXXXXX G     SCALER=  0040        SEMITG= XXXXXX G     SETARG= XXXXXX G
SETERA= XXXXXX G     SETERR= XXXXXX G     SIZCOD= XXXXXX G     STAX  = XXXXXX G     STOP     0038RG
STPFLG=  0040        STPTR = XXXXXX G     STRING=  0010        TRCFLG=  0020        TRUTH = XXXXXX G
TYPARG= XXXXXX G     T4    = XXXXXX G     UNDEF =  0080        VALERR=  0040        VALTG = XXXXXX G
ZX    = XXXXXX G
  ABS.    0000      00
          01C8      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2498. WORDS
.SY:FLCTRL/C:DK1:SEICL,FLCTRL
```

CROSS REFERENCE TABLE (CREF V01-03)

2-5

A11X 7-11#
A5X 7-11# 15-36 15-43
A6X 7-11#
A7X 7-11# 15-39 15-47
A8X 7-11#
A9X 7-11# 15-48
ABRFLG 2-10#
AFAIL 5-38# 14-28
ALLOK 4-13#
ALLTG 5-24#
AMAT 5-37#
ARRAY 4-10#
ASGCOD 1-36#
ASTR 5-36#
ATSNTG 5-22#
BACKUP 7-6# 15-28 15-72
BAKSTG 5-23#
BMAT 5-40#
BSTR 5-39#
CALLTG 5-27#
CDQPTR 3-40#
CDSPTR 3-39#
CLOSE 7-8#
CLPTR 2-4# 10-16 10-17 10-18 12-30* 12-32* 12-33 14-18 14-32 14-37 14-39 15-66*
CLRARG 7-7# 11-19
CMAT 5-42#
CONCOD 1-45#
CRCOD 1-44#
CSTR 5-41#
CTKN 2-7#
D8X 7-11# 11-23
D9X 7-11# 11-17
DATCOD 1-47#
DIMFLG 2-13#
DISSRQ 2-21#
DREXTA 2-25# 7-42 9-15 10-31 11-52
DREXTB 2-26#
END 7-8# 12-39
EOFCOD 1-41#
EOFTBL 3-5#
EOLTG 5-28# 7-53 10-29 11-14
EOSTG 5-29# 7-45
EQUCOD 1-37#
ERASGN 6-16#
ERBRK 6-14#
ERDOMN 6-7#
EREOFN 6-15#
ERFORA 6-11# 14-46
ERLNNF 6-8#
ERNOT 6-13#
ERNIMX 6-5# 14-16
ERNOFN 6-4#
ERNXFN 6-20#
ERNXTA 6-12# 15-26
EROFA 6-10# 11-58

CROSS REFERENCE TABLE (CREF V01-03)

```
ERRCD   2-31#   7-31    9-11    10-1    11-54#  13-12
ERRCDS  2-32#
ERSHAP  6-18#
ERSTOP  6-9#    8-8
ERUNDF  6-17#
ERVAL   6-19#
ERUSEL  6-6#
ESTG    5-6#    12-14
EXTFLG  2-9#
FIX1    7-9#    11-24
FIXUP   10-27#  14-43
FLCTRL  7-4#    7-5#
FNARGD  1-38#
FNFLG   2-12#
FNTBL   3-6#
FOR     14-2#   14-12#
FORERR  14-22   14-29   14-45#
FOROK   14-14   14-18#
FORPV   14-27   14-32#
FORTG   5-8#    14-41   15-29
FPADD   7-9#    15-41
FPONE   7-10#   14-30
FPSUB   7-9#
FUZZIE  7-9#    15-50
GETLN   7-6#    7-30    9-10    10-11   13-11
GLBFLG  2-16#   7-36    7-37#   7-41#   14-12
GOSEXT  10-13   10-31#
GOSSKP  10-20   10-23#
GOSTAG  10-27   10-29#
GOSTG   5-7#    10-25   12-15   12-20
GOSUB   10-2#   10-11#
GOTEXT  9-12    9-15#
GOTO    9-2#    9-10#
HALTA   7-12#   11-55   12-37   13-18   15-70
IDLE    7-8#    7-51    10-28
IF      13-2#   13-11#
IFABR   13-13   13-15   13-18#
IMACOD  1-48#
IMXFLG  2-11#
IMXTG   5-11#
ITM1TG  5-18#
ITM2TG  5-19#
JMPABX  3-35#
JMPX    3-19#
KEYFLG  2-20#   7-49
KEYSTK  7-7#
LBRKTG  5-20#
LCLFLG  2-8#
LDBX    3-20#
LDBX    3-25#
LDXX    3-15#
LISTTG  5-9#
LITCOD  1-46#
LNNOTG  5-17#
LOCTG   7-6#    11-15   15-30
LOCTGR  7-6#    12-16
LSP     2-48#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
LSTCOD  1-49#
MMICOD  1-43#
MMLCOD  1-42#
MULCOD  1-40#
NCSFLG  2-23#
NEXT    15-2#   15-17#
NLPTR   2-5#    7-36#   9-14#   10-15#  12-36#  13-17#  15-69#
NTAPTR  4-20#
NTATTR  4-7#    14-20   15-21
NTDIMS  4-19#
NTDLEN  4-22#
NTLINK  4-4#
NTNAME  4-5#
NTPTR   2-6#
NTSPTR  4-24#
NTVCL   4-15#
NTWCOL  4-18#
NTWLEN  4-23#
NTWROW  4-17#
NULLTG  5-4#
NXFOR   15-12#  15-13   15-65
NXINCR  15-14#  15-15   15-52
NXNT    15-13#  15-14   15-33
NXSTGP  15-15#
NXTBBR  15-70#  15-74
NXTAG   15-11#  15-12
NXTBCK  15-53   15-58#
NXTEND  15-55   15-59   15-72#
NXTERR  15-25#  15-32
NXTEXT  15-67   15-69#
NXTLP   15-28#  15-35
NXTOK   15-24   15-29#
NXTSKP  15-56   15-61#
OBJGTR  4-30#
OBJBCK  4-31#
OBJPT   4-32#
OBJLEN  4-28#
OF      11-2#   11-13#
OFABR   11-27   11-29   11-46   11-55#
OFERR   11-21   11-57#
OFEXIT  11-52#
OFRST   11-25   11-54#
OMSFLG  2-14#
ONTBL   3-4#
OPRADR  2-24#
OVAFLG  2-22#
PAETG   5-15#
PARM    4-12#
PGMSTR  4-37#
PGMBP   4-39#
PGMCD   4-41#   14-19   14-33   14-35
PGMFP   4-38#   12-35   15-68
PGMLEN  4-36#
PGMLNN  4-40#   10-19
PGMPTP  2-50#   7-48
PGMTG   5-10#
PLOSTG  5-5#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

PLRB 1-20#
PLRB 1-27#
SET 1-3#

```
FFFFFFFFFF  LL          DDDDDDDDD    AAAAAAAA   TTTTTTTTTT   AAAAAAAA          LL          SSSSSSSS   TTTTTTTTTT
FFFFFFFFFF  LL          DDDDDDDDDD  AAAAAAAAAA  TTTTTTTTTT  AAAAAAAAAA         LL          SSSSSSSSSS  TTTTTTTTTT
FF          LL          DD      DD  AA      AA      TT      AA      AA         LL          SS      S       TT
FF          LL          DD      DD  AA      AA      TT      AA      AA         LL          SS              TT
FFFFFF      LL          DD      DD  AA      AA      TT      AA      AA         LL          SSSSSSSS        TT
FFFFFF      LL          DD      DD  AAAAAAAAAA      TT      AAAAAAAAAA         LL          SSSSSSSS        TT
FF          LL          DD      DD  AAAAAAAAAA      TT      AAAAAAAAAA         LL                SS        TT
FF          LL          DD      DD  AA      AA      TT      AA      AA         LL          S     SS        TT
FF          LLLLLLLLLL  DDDDDDDDD   AA      AA      TT      AA      AA  .....  LLLLLLLLLL  SSSSSSSSS     TT
FF          LLLLLLLLLL  DDDDDDDDD   AA      AA      TT      AA      AA  ....   LLLLLLLLLL  SSSSSSSS      TT
```

TABLE OF CONTENTS

```
                                        .TITLE  FLDATA FLOW OF DATA
                                        .IDENT  /BE2021/
                                ;
                                        .GLOBL  FLDATA
                        0000'   FLDATA  =       .
                                        .GLOBL  LOCTG,TYPARG,SETARG,TYPRES,BACKUP,GETLN
                                        .GLOBL  PULFPN,PSHFPN,FLOAT1,A5X,A8X,A10X,ZX
                                        .GLOBL  SYSERR,SETERR,SETERR
                                        .GLOBL  ITGCOD,FIX1
                                ;
                                        .SBTTL  XXX DATACN      DATA CONSTANT IN OBJECT STRING
                                        .GLOBL  DATACN
                                ;
                                ;       PUSH THE CONSTANT IN THE OBJECT STRING ONTO THE STACK
                                ;       ALSO UPDATE NTPTR
                                ;
0000    DE      00G     DATACN: LDX     NTPTR,D         ;GET PTR TO FIRST BYTE OF DATA
0002    BD      0000G           JSR     PSHFPN          ;MOVE IT ONTO THE STACK
0005    4F                      CLR A                   ;UPDATE TOKEN POINTER
0006    C6      08              LDA B   #.I
0008    DB      01G     ADDNT:  ADD B   NTPTR+1,D
000A    99      00G             ADC A   NTPTR,D
000C    97      00G             STA A   NTPTR,D
000E    D7      01G             STA B   NTPTR+1,D
0010    7E      0000G           JMP     DREXTA
```

```
                                        .SBTTL  XXX LITCN       LITERAL CONSTANT IN OBJECT STRING
                                        .GLOBL  LITCN

                                ;
                                ;       PUSH PTR ON STACK FOR LITERAL IN STRING
                                ;       ALSO UPDATE NTPTR
                                ;
0013    96      01G     LITCN:  LDA A   NTPTR+1,D       ;PUSH PTR TO STRING COUNT
0015    36                      PSH A
0016    96      00G             LDA A   NTPTR,D
0018    36                      PSH A
0019    96      01G             LDA A   CLPTR+1,D       ;PUSH CLPTR
001B    36                      PSH A
001C    96      00G             LDA A   CLPTR,D
001E    36                      PSH A
001F    86      00G             LDA A   PLOSTG,I        ;TAG STRING ENTRY
0021    36                      PSH A
0022    DE      00G             LDX     NTPTR,D         ;GET COUNT FOR NTPTR UPDATE
0024    A6      00              LDA A   0,X
0026    E6      01              LDA B   1,X
0028    CB      02              ADD B   2,I             ;ADD SPACE FOR COUNT
002A    89      00              ADC A   0,I
002C    20      DA              BRA     ADDNT           ;JOIN COMMON UPDATE CODE
```

```
                                        .SBTTL  XXX INTCN       INTEGER IN THE OBJECT LINE
                                        .GLOBL  INTCN
                                ;
                                ;       PUSH INTEGER AND FLOAT IT
                                ;
002E    DE      00G     INTCN:  LDX     NTPTR,D         ;GET NEXT OBJECT BYTE ADDR
0030    A6      01              LDA A   1,X
0032    36                      PSH A
0033    A6      00              LDA A   0,X
0035    36                      PSH A
0036    86      00G             LDA A   #TMITG.I
0038    36                      PSH A
0039    08                      INX                     ;SKIP DATA IN OBJECT LINE
003A    08                      INX
003B    DF      00G             STX     NTPTR,D
003D    BD      0000G           JSR     FLOAT1          ;FLOAT IT
0040    7E      0000G           JMP     DREXTA          ;SO LONG
                                .SBTTL  XXX LINCN       LINE NUMBER
                                .GLOBL  LINCN
                                ;
                                ;       LINE NUMBER IS REFERANCED IN OBJECT LINE
                                ;
002E'                   LINCN   =       INTCN           ;DUPLICATE CODE FOR NOW
```

```
                                        .SBTTL  *** PTRNTN      POINTER TO NAME TABLE NUMERIC
                                        .GLOBL  PTRNTN
                                ;
                                ;       PUSH PTR ON STACK FOR NAME TABLE NUMERIC OBJECT ENTRY
                                ;       ALSO UPDATE NTPTR
                                ;
                                ;       NOTE.. SPACE IS PROVIDED SO THE ENTRY CAN BE CONVERTED TO A VALUE
                                ;
0043    36              PTRNTN: PSH A
0044    36                      PSH A
0045    36                      PSH A
0046    36                      PSH A
0047    36                      PSH A
0048    36                      PSH A
0049    DE      000G            LDX     NTPTR,D         ;GET PTR TO PTR
0048    A6      01              LDA A   1,X             ;PUSH PTR TO NT ON STACK
004D    36                      PSH A
004E    A6      00              LDA A   0,X
0050    36                      PSH A
0051    86      000G            LDA A   PNTNTG.I        ;TAG ENTRY
0053    36                      PSH A
0054    08              SKIP:   INX                     ;UPDATE NTPTR
0055    08                      INX
0056    DF      000G            STX     NTPTR,D
0058    7E      0000G           JMP     DREXTA
```

```
                                        .SBTTL  XXX PTRNTS      POINTER TO NAME TABLE STRING
                                        .GLOBL  PTRNTS
                                ;
                                ;       PUSH POINTER ON STACK FOR STRING NAME TABLE ENTRY
                                ;       ALSO UPDATE NTPTR
                                ;
0058    4F              PTRNTS: CLR A                   ;PUSH 2 FREE BYTES ON STACK
005C    36                      PSH A
0050    36                      PSH A
005E    DE      006             LDX     NTPTR,D         ;GET PTR TO NAME TABLE
0060    A6      01              LDA A   1,X
0062    36                      PSH A
0063    A6      00              LDA A   0,X
0065    36                      PSH A
0066    B6      006             LDA A   PNTSTG.1        ;TAG ENTRY
0068    36                      PSH A
0069    20      E9              BRA     SKIP            ;GO UPDATE NTPTR
```

```
                                        .SBTTL  XXX FNPARM      PARAMETER FOR DEF STATEMENT
                                        .GLOBL  FNPARM
                                ;
                                ;       FIND VALUE THAT WAS PARAMETER FOR FUNCTION AND PUSH IT FOR CALLER
                                ;
                                ;       STACK FORMAT FOR PARAMETER
                                ;
                                ;       ITEM            LENGTH IN BYTES
                                ;       VALUE           9               (IT MUST BE A REAL VALUE BY NOW)
                                ;       RTRN ADDR       3
                                ;       EVAL STATUS     9
                                ;       EOL/RTRN ADDR   3
                                ;
                        000F    FNSIZE  =       15.             ;BYTES TO SKIP FROM EOL
                                ;
0068    9F      00G     FNPARM: STS     R0,D            ;SET UP FOR CALLS
006D    86      00G             LDA A   EOLTG,I         ;PARM DATA IS BEFORE LINE START
006F    BD      0000G           JSR     LOCTG
0072    DE      00G             LDX     R0,D            ;IF ZERO FAILED
0074    27      12              BEQ     PARMER
0076    86      09              LDA A   9.,I            ;BYTES TO MOVE TO BOTTOM OF STACK
0078    E6      18      PARMLP: LDA B   FNSIZE+9.,X
007A    37                      PSH B
007B    09                      DEX
007C    4A                      DEC A
007D    26      F9              BNE     PARMLP
007F    DE      00G             LDX     NTPTR,D         ;SKIP PTR IN OBJECT LINE
0081    08                      INX
0082    08                      INX
0083    DF      00G             STX     NTPTR,D
0085    7E      0000G           JMP     DREXTA          ;ALL READY
                                ;
0088    BD      0000G   PARMER: JSR     SYSERR
```

```
                                        .SBTTL  *** FNASGN      DEF FUNCTION ASSIGN
                                        .GLOBL  FNASGN
                                ;
                                ;       TAKE VALUE ON STACK AND MOVE IT OVER OLD CALLERS VALUE.
                                ;
0088    9F      00G     FNASGN: STS     R0,D            ;SET UP FOR CALLS
008D    86      00G             LDA A   EQLTG,I
008F    BD      0000G           JSR     LOCTG           ;FIND END OF LINE
0092    DE      00G             LDX     R0,D            ;IF ZERO I FAILED
0094    27      13              BEQ     FNASER
0096    31                      INS                     ;MOVE VALUE TO AREA FOR CALLER TO USE
0097    86      08              LDA A   8.,I
0099    33              FNASLP: PUL B
009A    E7      11              STA B   FNSIZE+2,X
009C    08                      INX
009D    4A                      DEC A
009E    26      F9              BNE     FNASLP
00A0    86      C0              LDA A   ABRFLG+EXTFLG,I ;THIS IS END OF DEF FUNCTION,
00A2    9A      00G             ORA A   LCLFLG,D        ;  SO TERMINATE LINE EXEC AND GO
00A4    97      00G             STA A   LCLFLG,D        ;  BACK TO CALLING LINE
00A6    7E      0000G           JMP     DREXTA
                                ;
00A9    BD      0000G   FNASER: JSR     SYSERR
```

```
                                                .SBTTL  XXX GTDATA      GET DATA OBJECT FROM DATA STMT
                                                .GLOBL  GTDATA
                                        ;
                                        ;       PRUNE ONE OBJECT OFF CURRENT DATA STATEMENT AND RETURN POINTERS TO IT.
                                        ;       IF THERE IS NO ACTIVE STATEMENT FIND ONE.
                                        ;       IF THE CURRENT STATEMENT IS EXAUSTED FIND A NEW ONE
                                        ;       IF THERE ARE NO MORE DATA STATEMENTS SET ERROR CODE
                                        ;
00AC    FE    0000G     GTDATA: LDX     CDOPTR          ;IS THERE AN OBJECT
00AF    27    33                BEQ     GTDNLA          ;IF%0 NEED TO START WITH NEW DATA STMT
00B1    4F              GTDLPA: CLR A                   ;HIGH HALF OF BYTE COUNT IS USUALY ZERO
00B2    E6    00                LDA B   0,X             ;TEST TYPE OF CURRENT OBJECT
00B4    C1    00G               CMP B   CRCOD,I         ;IS IT END OF LINE
00B6    27    41                BEQ     GTDNLC
00B8    C1    00G               CMP B   LITCOD,I        ;LITERAL IN STRING
00BA    27    20                BEQ     GTDLIT
00BC    C1    00G               CMP B   CONCOD,I        ;IS IT A VALUE
00BE    27    18                BEQ     GTDCON
00C0    C1    00G               CMP B   ITGCOD,I        ;IS IT AN INTEGER
00C2    27    03                BEQ     GTDITG
00C4    BD    0000G             JSR     SYSERR          ;CURSE YOU
                                ;
00C7    C6    03        GTDITG: LDA B   3,I             ;LENGTH OF DATA AND TAG
00C9    FB    0001G     GTDEXT: ADD B   CDOPTR+1        ;GET NEXT OBJECT ADDR
00CC    B9    0000G             ADC A   CDOPTR
00CF    B7    0000G             STA A   CDOPTR
00D2    F7    0001G             STA B   CDOPTR+1
00D5    DF    00G               STX     R0,D
00D7    39                      RTS
                                ;
00D8    C6    09        GTDCON: LDA B   9.,I            ;UPDATE CDOPTR
00DA    20    ED                BRA     GTDEXT
                                ;
00DC    C6    03        GTDLIT: LDA B   3,I             ;LENGTH OF AG AND COUNT
00DE    EB    02                ADD B   2,X
00E0    A9    01                ADC A   1,X
00E2    20    E5                BRA     GTDEXT
                                ;
00E4    DE    00G       GTDNLA: LDX     PGMPTR,D        ;IF NO PROGRAM DIE
00E6    27    18                BEQ     GTDERR
00E8    FF    0000G     GTDNLB: STX     CDSPTR
00EB    A6    09                LDA A   PGMCD,X         ;DATA STMT HAS DATCOD IN FIRST BYTE
00ED    81    00G               CMP A   DATCOD,I
]]]]]
00EF    26    08                BNE     GTDNLD
00F1    BD    0000G             JSR     A10X            ;SET UP OBJECY POINTER
00F4    FF    0000G             STX     CDOPTR
00F7    20    B8                BRA     GTDLPA
                                ;
00F9    FE    0000G     GTDNLC: LDX     CDSPTR          ;GET ADDR OF CURRENT DATA STATEMENT
00FC    EE    03        GTDNLD: LDX     PGMFP,X         ;TRY NEXT LINE FOR DATA STMT
00FE    26    E8                BNE     GTDNLB
                                ;
0100    8D    04        GTDERR: BSR     RESTZ
0102    8D    0000G             JSR     SETERR          ;SET ERROR CODE
0105    00G                     .BYTE   ERNDT
```

```
                                        .SBTTL  XXX RESTORE     RESTORE STATEMENT
                                        .GLOBL  RESTZ,RESTS
                                ;
                                ;       SET UP CURRENT DATA STATEMENT POINTERS SO GTDATA WILL START OVER
                                ;
0106    0E      000     RESTZ:  LDX     ZX,D            ;IF CDSPTR & CDOPTR = 0 THEN DATA IS
0108    FF      00000G          STX     CDSPTR          ;  RESET AND CAN START AT FIRST LINE
010B    FF      00000G          STX     CDOPTR
010E    39                      RTS
                                ;
                                ;       IMPLEMENT RESTORE COMMAND
                                ;       JUST RESET POINTERS IF NO VALUE ON THE STACK
                                ;       IF THERE IS A VALUE ON THE STACK USE IT AS A LINE POINTER,
                                ;         IF IT IS A DATA STATEMENT SET UP POINTERS TO START THERE
                                ;         IF NOT A DATA STATEMENT SET ERROR AND EXIT.
                                ;
010F    8D      F5      RESTS:  BSR     RESTZ           ;PRESET PTR
0111    30                      TSX                     ;SEE IF I GOT A LINE NUMBER
0112    A6      00              LDA A   0,X
0114    81      000             CMP A   VALTG.I
0116    26      16              BNE     RSTNLN
0118    8D      00000G          JSR     GETLN           ;FIND FUZZIE LINE
011B    DE      C0G             LDX     R0.D
011D    27      0F              BEQ     RSTNLN          ;IT FAILED TO FIND LINE
011F    86      000             LDA A   DATCOD.I        ;DATA STATEMENT IS NEEDED
0121    A1      09              CMP A   PGMCD.X
0123    26      0C              BNE     RSTFL           ;NOT DATA STMT
0125    FF      00000G          STX     CDSPTR          ;STATEMENT ADDR
0128    8D      00000G          JSR     A10X            ;GET FIRST ITEM ADDR
012B    FF      00000G          STX     CDOPTR          ;OBJECT POINTER
012E    7E      00000G  RSTNLN: JMP     DREXTA
                                ;
0131    8D      00000G  RSTFL:  JSR     SETERA          ;OOPS
0134    00G                     .BYTE   ERNDT
```

```
                                        .SBTTL  XXX ASG/MATMOV ROUTINES
                                        .GLOBL  ASG.MATMOV.ASSCSC
                                ;
                                ;
                                ;       INPUTS:
                                ;
                                ;               RESULT AREA POINTER (NAME TABLE  OR POINTER TO
                                ;                       ARRAY ELEMENT)
                                ;               INPUT DATA OR POINTER
                                ;
                                ;       INPUT IS TESTED FOR VALIDITY AND MOVED TO RESULT AREA
                                ;
                                ;               ASMM - ASSIGN OF A MATRIX FROM A MATRIX
                                ;               ASMSC- ASSIGN OF A MATRIX FROM A SCALAR
                                ;               ASSTST-ASSIGN OF A STRING FROM A STRING
                                ;               ASSCSC-ASSIGN OF A SCALAR FROM A SCALAR
                                ;
                                ;       CODE IS IN FUNNY ORDER SO BRANCHES WORK
                                ;
                                ; REGISTER USAGE IN   --MATRIX RECEIVES MATRIX--
                                ;
                                ;       R0 - STACK POINTER FOR PRUNE
                                ;       R1 - INPUT DATA ADDR-5 BYTES
                                ;       R2 - INPUT RESET ADDR FOR MATRIX (- SCALER
                                ;            0 FOR MATRIX (- MATRIX
                                ;       R3 - OUTPUT DATA ADDR-
                                ;       R4 - END OF OUTPUT FIELD-5
                                ;
                                ;       MATMOV IS FOR MAT INV TO USE AS UTILITY
                                ;
0135    9F      000     MATMOV: STS     R0,D            ;DON'T REALLY PRUNE STACK
                        ;
0137    30              ASMM:   TSX                     ;FIND SHAPE & SIZE OUTPUT MATRIX
0138    A6      05              LDA A   5,X             ;TEST FOR CONFORMABLE SHAPES
013A    A1      0E              CMP A   14.,X
013C    26      63              BNE     ASERRA
013E    A6      06              LDA A   6,X
0140    A1      0F              CMP A   15.,X
0142    26      50              BNE     ASERRA
0144    A6      07              LDA A   7,X
0146    A1      10              CMP A   16.,X
0148    26      57              BNE     ASERRA
014A    A6      08              LDA A   8.,X
014C    A1      11              CMP A   17.,X
014E    26      54              BNE     ASERRA
0150    EE      01              LDX     1,X             ;GET INPUT DATA ADDR
0152    EE      08              LDX     NTAPTR,X
0154    BD      0000G           JSR     A5X
0157    DF      000             STX     R1,D
0159    CE      0000            LDX     0,I
015C    DF      000             STX     R2,D            ;CLEAR FOR NO RESET OF INPUT ADDRESSES
                        ;
015E    30              ASMCOM: TSX                     ,GET COMMON OUTPUT PARMS
015F    EE      0A              LDX     10.,X
0161    A6      04              LDA A   NTATTR,X        ;DATA WILL BE THERE
```

```
 58  0163  8A  04          ORA A   ALLOK,I
 59  0165  A7  04          STA A   NTATTR,X
 60  0167  EE  08          LDX     NTAPTR,X
 61  0169  BD  0000G       JSR     ASX
 62  016C  DF  00G         STX     R3,D
 63  016E  30              TSX
 64  016F  EE  0C          LDX     12.,X
 65  0171  BD  0000G       JSR     ASX
 66  0174  DF  00G         STX     R4,D
 67                        ;
 68  0176  DE  00G         LDX     R1,D
 69  0178  9C  00G         CPX     R3,D
 70  017A  27  20          BEQ     ASMMEX
 71                        ;
 72  017C  DE  00G         ASMLP:  LDX     R1,D
 73  017E  BD  0000G       JSR     PSHFPN
 74  0181  BD  0000G       JSR     A8X
 75  0184  DF  00G         STX     R1,D
 76  0186  DE  00G         LDX     R3,D
 77  0188  BD  0000G       JSR     PULFPN
 78  0188  BD  0000G       JSR     A8X
 79  018E  DF  00G         STX     R3,D
 80  0190  DE  00G         LDX     R2,D            ; IN ASMM OR ASMSC?
 81  0192  27  02          BEQ     ASMSKP          ; BRANCH IF ASMM
 82  0194  DF  00G         STX     R1,D
 83  0196  DE  00G         ASMSKP: LDX     R3,D    ;IS IT TIME TO STOP
 84  0198  9C  00G         CPX     R4,D
 85  019A  26  E0          BNE     ASMLP
 86  019C  9E  00G         ASMMEX: LDS     R0,D    ; PRUNE STACK AND EXIT.
 87  019E  7E  0000G       JMP     DREXTA
 88                        ;
 89  01A1  BD  0000G       ASERRA: JSR     SETERA  ;SET ERROR CODE AND EXIT
 90  01A4  00G             .BYTE   ERASGN
 91                        ;
 92                        ;****** NOW DO THE SETUP TO USE CODE IN ASMM TO ACCOMPLISH THE ASMSC***
 93                        ;       THIS IS A GOOD EXAMPLE OF DIRTY CODE!!!!!
 94                        ;
 95  01A5  30              ASMSC:  TSX             ; GET
 96  01A6  08                      INX
 97  01A7  DF  00G                 STX     R1,D    ; INPUT ADDRESS.
 98  01A9  DF  00G                 STX     R2,D    ; ASMSC ACTIVE FLAG AND RESET.
 99  01AB  20  B1                  BRA     ASMCOM  ; STRUCTURED PROGRAMMERS, EAT YOUR HEARTS
100                                               ;   OUT.  (END OF DIRTY CODE)
101                       ;
102                       ;       ***************************
103                       ;       ***                     ***
104                       ;       ***  MAIN ENTRY POINT   ***
105                       ;       ***                     ***
106                       ;       ***************************
107                       ;
108 01AD  BD  0000G       ASG:    JSR     SETARG  ;TEST AND PREPARE INPUT AREA
109 0180  96  00G                 LDA A   ERRCD,D ;DID IT WORK
110 0182  26  3D                  BNE     ASKILL  ; IF CODE IS SET EXIT NOW
111 0184  DE  00G                 LDX     NTPTR,D ;MUST HAVE ASSIGN IN LINE
112 0186  09                      DEX
113 0187  DF  00G                 STX     NTPTR,D
114 0189  BD  0000G               JSR     TYPRES  ;SAME FOR RESULT.
```

```
        01BC    96    00G                LDA A   ERRCD,D        ;DID IT BLOW UP
        01BE    26    31                 BNE     ASKILL
        01C0    BD    0000G              JSR     BACKUP         ;R0 NOW HAS ADDRESS OF PURNED STACK.
        01C3    96    00G                LDA A   R4,D
        01C5    27    13                 BEQ     ASSCSC         ;SCALER RECEIVES SCALER.
        01C7    81    A0                 CMP A   RSTR+ASTR,I    ; STRING RECEIVES STRING.
        01C9    27    2D                 BEQ     ASSTST
        01CB    81    40                 CMP A   RMAT,I         ; MATRIX RECEIVES SCALER.
        01CD    27    D6                 BEQ     ASMSC
        01CF    81    50                 CMP A   RMAT+AMAT,I    ;MATRIX RECEIVES MATRIX.
        01D1    26    03                 BNE     ASERRB
        01D3    7E    0137'              JMP     ASMM           ;BRANCH ADDRESSING WON'T MAKE IT
                                 ;
        01D6    BD    0000G      ASERRB: JSR     SETERA
        01D9    00G                      .BYTE   ERASGN
                                 ;
                                 ;
        01DA    30               ASSCSC: TSX                    ; WANT TO TEST TAG.
        01DB    A6    09                 LDA A   9.,X
        01DD    81    00G                CMP A   PAETG,I        ;
        01DF    27    13                 BEQ     ASPAE          ;
        01E1    EE    0A                 LDX     10.,X          ; GET PTR TO NAME TABLE
        01E3    A6    04                 LDA A   NTATTR,X       ;MARK AS DEFINED EVEN IF IT WAS
        01E5    84    7F                 AND A   255.-UNDEF,I
        01E7    A7    04                 STA A   NTATTR,X
        01E9    BD    0000G              JSR     A5X            ; SKIP NAME TABLE HEADER STUFF
        01EC    BD    0000G      ASSSCM: JSR     PULFPN         ; MOVE FROM STACK TO RAM
        01EF    9E    00G        ASEXIT: LDS     R0,D           ; PRUNE STACK.
        01F1    7E    0000G      ASKILL: JMP     DREXTA         ; BYE BYE FOLKS!!!!!!!
                                 ;
        01F4    EE    0C         ASPAE:  LDX     12.,X          ; GET POINTER TO DATA
        01F6    20    F4                 BRA     ASSSCM         ; GO TO COMMON CODE
                                 ;
                                 ;
                                 ; XXXXXX STRING RECEIVES STRING REGISTER USAGE XXXXX
                                 ;       R0 - DON'T USE IT!!!!!!
                                 ;       R1 - INPUT ADDRESS.
                                 ;       R2 - OUTPUT ADDRESS.
                                 ;       R3 - ADDRESS OF LIST OUTPUT BYTE+1
                                 ;
        01F8    30               ASSTST: TSX                    ; GET INPUT JUNK.
        01F9    EE    03                 LDX     3,X
        01FB    A6    00                 LDA A   0,X            ; GET
        01FD    E6    01                 LDA B   1,X            ;   THE
        01FF    DF    00G                STX     R1,D           ;   ADDRESS.
        0201    30                       TSX                    ; NOW OUTPUT JUNK.
        0202    EE    06                 LDX     6,X
        0204    A1    05                 CMP A   NTDLEN,X       ; WILL IT FIT?
        0206    22    CE                 BHI     ASERRB         ; TRANSFER IF THE NEW LENGTH IS
                                                                ;  GREATER THAN THE DIMENSIONED LENGTH.
        0208    25    04                 BCS     ASOKJ          ; TRANSFER IF NEW << DIMENSIONED
        020A    E1    06                 CMP B   NTDLEN+1,X     ; IF NOT RESOLVED IN FIRST BYTE.
        020C    22    C8                 BHI     ASERRB
        020E                     ASOKJ:  SEI
   020E 01    0F                         .BYTE   01,17
        0210    97    00G                STA A   R4,D
        0212    D7    01G                STA B   R4+1,D
```

```
      0214   DF   00G            STX    R5,D           ;SAVE NT ADDR
      0216   6F   07             CLR    NTWLEN,X       ;SET LEN TO ZERO FOR ABORT
      0218   6F   08             CLR    NTWLEN+1,X
      021A   EB   0C             ADD B  NTSPTR+1,X     ; CLAC END OF DATA ADDR
      021C   A9   0B             ADC A  NTSPTR,X
      021E   97   00G            STA A  R3,D
      0220   D7   01G            STA B  R3+1,D
      0222   EE   0B             LDX    NTSPTR,X       ; GET DATA
      0224   DF   00G            STX    R2,D           ;   OBJECT POINTER.
      0226   0E                  CLI
      0227   9C   00G            CPX    R3,D           ;IS IT A NULL STRING
      0229   27   12             BEQ    ASENDS         ;IF SO STOP NOW
                         ;
                         ;** THERE IS NO NICE WAY TO MOVE DATA. SO HERE IS BRUTE FORCE METHOD**
                         ;
      022B   DE   00G    ASLPJ:  LDX    R1,D           ; *** AND
      022D   E6   02             LDA B  2,X            ;
      022F   08                  INX                   ;
      0230   DF   00G            STX    R1,D           ; *** AWAY
      0232   DE   00G            LDX    R2,D           ;
      0234   E7   05             STA B  OBJDT,X        ;
      0236   08                  INX
      0237   DF   00G            STX    R2,D           ; *** IT
      0239   9C   00G            CPX    R3,D           ;WAS IT LAST BYTE
      023B   26   EE             BNE    ASLPJ          ;
      023D   DE   00G    ASENDS: LDX    R5,D           ;GET NT PTR
      023F   96   00G            LDA A  R4,D           ;GET NEW WORKING LENGTH
      0241   D6   01G            LDA B  R4+1,D
      0243                       SEI
  0243   01   0F         .BYTE   01,17
      0245   A7   07             STA A  NTWLEN,X       ;FIX NT
      0247   E7   08             STA B  NTWLEN+1,X
      0249   86   10             LDA A  STRING,I       ;MARK AS DEFINED
      024B   A7   04             STA A  NTATTR,X
      024D   EE   0B             LDX    NTSPTR,X       ;GET OBJECT ADDR
      024F   86   20             LDA A  32.,I          ;MAK AS DEFINED
      0251   A7   02             STA A  OBJATR,X
      0253   0E                  CLI
      0254   20   99             BRA    ASEXIT         ; *** MOVED.
                         ;
             0001'               END
```

```
FLDATA FLOW OF DATA    RT-11 MMAC  VM02-10   14-OCT-76 01:32:47 PAGE 16+
SYMBOL TABLE

ABRFLG=  0040      ADDNT    0008R     AFAIL =  0030      ALLOK =  0004      ALLTG = 000000 G
AMAT  =  0010      ARRAY =  0020      ASENDS   0230R     ASERRA   0104R     ASERRB   0106R
ASEXIT   01EFR     ASG      01A0RG    ASGCOD= 000000 G   ASKILL   01F1R     ASLPJ    0228R
ASMCOM   015ER     ASMLP    017CR     ASMM     0137R     ASMMEX   019CR     ASMSC    01A5R
ASMSKP   0196R     ASOKJ    020ER     ASPAE    01E4R     ASSCSC   0100RG    ASSSCM   01ECR
ASSTST   01F8R     ASTR  =  0020      ATSNTG= 000000 G   A10X  = 000000 G   A8X   = 000000 G
A8X   = 000000 G   BACKUP= 000000 G   BAKSTG= 000000 G   BMAT  =  0004      BSTR  =  0008
CALLTG= 000000 G   CDOPTR= 000000 G   CDSPTR= 000000 G   CLPTR = 000000 G   CMAT  =  0001
CONCOD= 000000 G   CRCOD = 000000 G   CSTR  =  0002      CTKN  = 000000 G   DATACN   0000RG
DATCOD= 000000 G   DIMFLG=  0004      DISSRQ=  0008      DREXTA= 000000 G   DREXTB= 000000 G
EOFCOD= 000000 G   EOFTBL= 000000 G   EOLTG = 000000 G   EOSTG = 000000 G   EQUCOD= 000000 G
ERASGN= 000000 G   ERBRK = 000000 G   ERDOMN= 000000 G   EREOFN= 000000 G   ERFORA= 000000 G
ERLNNF= 000000 G   ERNOT = 000000 G   ERNIMX= 000000 G   ERNOFN= 000000 G   ERNOSN= 000000 G
ERNXTA= 000000 G   EROFA = 000000 G   ERRCD = 000000 G   ERRCOR= 000000 G   ERSHOR= 000000 G
ERSTOP= 000000 G   ERUNOF= 000000 G   ERVAL = 000000 G   ERWSFL= 000000 G   ESTG  = 000000 G
EXTFLG=  0080      FIX1  = 000000 G   FLDATA=  0000RG    FLOAT1= 000000 G   FNACOD= 000000 G
FNBSER   0099R     FNBSGN   0088RG    FNBSLP   0099R     FNELG =  0010      FNPARM   006BRG
FNSIZE=  000F      FNTBL = 000000 G   FORTG = 000000 G   GETLN = 000000 G   GLBFLG= 000000 G
GOSTG = 000000 G   GTDATA   00ACRG    GTDCON   0008R     GTDERR   0100R     GTDEXT   00C9R
GTDLTG   00C2R     GTDLIT   00DCR     GTDLPA   00B1R     GTDMLA   00E4R     GTDMLB   00E8R
GTDMLC   00F9R     GTDMLD   00FCR     IMACOD= 000000 G   IMDFLG=  0020      IMXTG = 000000 G
INTCN    002ERG    ITGCOD= 000000 G   ITM1TG= 000000 G   ITM2TG= 000000 G   JMPAX = 000000 G
JMPX  = 000000 G   KEYFLG=  0010      KEYSTK= 000000 G   LBRKTG= 000000 G   LCLFLG= 000000 G
LDAX  = 000000 G   LDBX  = 000000 G   LDXX  = 000000 G   LINCN =  002ERG    LISTTG= 000000 G
LITCN    0013RG    LITCOD= 000000 G   LNNOTG= 000000 G   LOCTG = 000000 G   LSP   = 000000 G
LSTCOD= 000000 G   MSGTMOV   0135RG    MMUCOD= 000000 G   MPLCOD= 000000 G   MULCOD= 000000 G
NCSFLG=  0001      NLPTR = 000000 G   NTAPTR=  0008      NTATTR=  0004      NTDIMS=  0009
NTDLEN=  0005      NTLINK=  0000      NTNAME=  0002      NTPTR = 000000 G   NTSPTR=  0008
NTVAL =  0005      NTWCOL=  0007      NTWLEN=  0007      NTWROW=  0006      NULLTG= 000000 G
OBJATR=  0002      OBJBCK=  0003      OBJOT =  0005      OBJLEN=  0000      ONSFLG=  0002
ONTBL = 000000 G   OPRADR= 000000 G   OUAFLG=  0002      PRETG = 000000 G   PARM  =  0008
PARMER   0088R     PARMLP   0079R     PGMATR=  0002      PGMBP =  0005      PGMCD =  0009
PGMFP =  0003      PGMLEN=  0000      PGMLMN=  0007      PGMPTR= 000000 G   PGMTG = 000000 G
PLOSTG= 000000 G   PNDEOF= 000000 G   PNDFLG= 000000 G   PNTNTG= 000000 G   PNTSTG= 000000 G
PRTTG = 000000 G   PSCTG = 000000 G   PSHEPN= 000000 G   PTRNTN   0043RG    PTRNTS   0058RG
PULFPN= 000000 G   RESTS    010FRG    RESTZ    0106RG    RFAIL =  00C0      RMAT  =  0040
RSTFL    0131R     RSTNLN   012ER     RSTR  =  0080      RTRNTG= 000000 G   RUNFLG=  008D
R0    = 000000 G   R1    = 000000 G   R10   = 000000 G   R11   = 000000 G   R2    = 000000 G
R3    = 000000 G   R4    = 000000 G   R5    = 000000 G   R6    = 000000 G   R7    = 000000 G
R8    = 000000 G   R9    = 000000 G   SBP   = 000000 G   SCALER=  0040      SEMITG= 000000 G
SETARG= 000000 G   SETERA= 000000 G   SETERR= 000000 G   SIZCOD= 000000 G   SKIP     0054R
STAX  = 000000 G   STPFLG=  0040      STPTR = 000000 G   STRING=  0010      SYSERR= 000000 G
TRCFLG=  0020      TYPARG= 000000 G   TYPRES= 000000 G   UNDEF =  0080      VALERR=  0040
VALTG = 000000 G   ZX    = 000000 G

  ABS.    0000    00
          0256    01

ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2518. WORDS.
.SY:FLDATA/C<DK1:SEICLI.FLDATA
```

CROSS REFERENCE TABLE (CREF V01-03)

```
        7-5
A10X    7-7#    14-45   15-29
A5X     7-7#    16-50   16-61   16-65   16-140
A8X     7-7#    16-74   16-75
ABRFLG  2-10#   13-18
ACONT   7-21#    8-22
ACALL   5-38#
ALLOK   4-13#   16-58
ALLTG   5-24#
AMAT    5-37#   16-124
ARRAY   4-10#
ASENDS  16-182  16-196#
ASERRA  16-38   16-41   16-44   16-47   16-89#
ASERRB  16-125  16-128#  16-163  16-167
ASEXIT  16-142#  16-208
ASG     16-2#   16-108#
ASGCOD   1-36#
ASKILL  16-110  16-116  16-143#
ASLPJ   16-186#  16-195
ASMLOM  16-55#  16-99
ASMLP   16-72#  16-85
ASMM    16-35#  16-126
ASMMEX  16-70   16-86#
ASMSC   16-95#  16-123
ASMSKP  16-81   16-87#
ASOKJ   16-165  16-168#
ASPAE   16-135  16-145#
ASSCSC  16-2#   16-119   16-132#
ASSSCM  16-141#  16-146
ASSTST  16-121  16-155#
ASTR    5-36#   16-120
ATSNTG  5-22#
BACKUP  7-6#    16-117
BAKSTG  5-23#
BMAT    5-40#
BSTR    5-39#
CALLTG  5-27#
CDOPTR  3-40#   14-9    14-24   14-25   14-26#  14-27#  14-46#  15-8#   15-30#
CDSPTR  3-39#   14-41#  14-49   15-7#   15-28#
CLPTR   2-4#     8-11    8-13
CMAT    5-42#
CONCOD  1-45#   14-17
CRCOD   1-44#   14-13
CSTR    5-41#
CTKN    2-7#
DATBCN  7-12#    7-17#
DATCOD  1-47#   14-43   15-25
DIMFLG  2-13#
DISSRQ  2-21#
DREXTA  2-25#    7-25    9-17   10-25   12-31   13-21   15-31   16-87   16-143
DREXTB  2-26#
EOFCOD  1-41#
EOFTBL  3-5#
EOLTG   5-28#   12-17   13-7
EOSTG   5-29#
EQUCOD  1-37#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
ERASGN  6-16#  16-90  16-129
ERBRK   6-14#
ERDOMN  6-7#
EREOFN  6-15#
ERFORA  6-11#
ERLMNF  6-8#
ERNOT   6-13#  14-56  15-74
ERNIMX  6-5#
ERNOFN  6-4#
ERNXFN  6-20#
ERNXTA  6-12#
EROFA   6-10#
ERRCS   2-31#  16-109  16-115
ERRCDB  2-32#
ERSHAP  6-18#
ERSTOP  6-9#
ERUNCF  6-17#
ERVAL   6-19#
ERWSFL  6-6#
ESTG    5-6#
EXTFLG  2-9#   13-18
FIX1    7-9#
FLDATA  7-4#    7-5#
FLOAT1  7-7#    9-16
FNACOD  1-38#
FNASER  13-10   13-23#
FNASGN  13-2#   13-6#
FNASLP  13-13#  13-17
FNFLG   2-12#
FNPARM  12-2#   12-16#
FNSIZE  12-14#  12-22  13-14#
FNTBL   3-6#
FORTG   5-8#
GETLN   7-6#    15-22
GLBFLG  2-16#
GOSTG   5-7#
GTDATA  14-2#   14-9#
GTDCON  14-18   14-31#
GTDERR  14-40   14-53#
GTDEXT  14-24#  14-32  14-37
GTDITG  14-20   14-23#
GTDLIT  14-16   14-34#
GTDLPB  14-11#  14-47
GTDNLA  14-10   14-39#
GTDNLB  14-41#  14-51
GTDNLC  14-14   14-49#
GTDNLD  14-44   14-50#
IMACOD  1-48#
IMXFLG  2-11#
IMXTG   5-11#
INTCN   9-2#    9-6#    9-23
ITGCOD  2-9#    14-19
ITM1TG  5-18#   9-11
ITM2TG  5-19#
JMPABX  3-35#
JMPX    3-10#
KEYFLG  2-20#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
PNTNTG  5-14#  10-20
PNTSTG  5-12#  11-15
PRTTG   5-26#
PSCTG   5-13#
PSHFPN  7-7#   7-18   16-73
PTRNTN  10-2#  10-9#
PTRNTS  11-2#  11-7#
PULFPN  7-7#   16-77  16-141
R0      2-28#  12-16* 12-19  13-6*  13-9   14-28* 15-23  16-33* 16-86  16-142
R1      2-28#  16-51* 16-68  16-72  16-75* 16-82* 16-97* 16-159* 16-186 16-189*
R10     2-29#
R11     2-29#
R2      2-28#  16-53* 16-80  16-98* 16-179* 16-190 16-193*
R3      2-28#  16-62* 16-69  16-76  16-79* 16-83  16-176* 16-177* 16-181 16-194
R4      2-28#  16-66* 16-84  16-118 16-169* 16-170* 16-197 16-198
R5      2-28#  16-121* 16-196
R6      2-28#
R7      2-28#
R8      2-29#
R9      2-29#
RESTS   15-2#  15-17#
RESTZ   14-53  15-2#  15-6#  15-12
RFAIL   5-35#
RMAT    5-34#  16-122 16-124
RSTEL   15-27  15-33#
RSTMLN  15-21  15-24  15-31#
RSTR    5-33#  16-120
RTRNTG  5-25#
RUNFLG  2-17#
SBP     2-49#
SCALER  4-9#
SEMITG  5-21#
SETARG  7-6#   16-108
SETERR  7-8#   15-33  16-89  16-128
SETERR  7-8#   14-54
SIZCOD  1-39#
SKIP    10-22# 11-17
STAX    3-30#
STPFLG  2-18#
STPTR   2-51#
STRING  4-11#  16-202
SYSERR  7-8#   12-33  13-23  14-21
TRCFLG  2-19#
TYPARG  7-6#
TYPRES  7-6#   16-114
UNDEF   4-8#   16-138
VALERR  4-29#
VALTG   5-16#  15-20
ZX      7-7#   15-6
```

CROSS REFERENCE TABLE (CREF V01-03)

PLRA 1-20#
PLRB 1-27#
SE1 1-3# 16-168 16-199

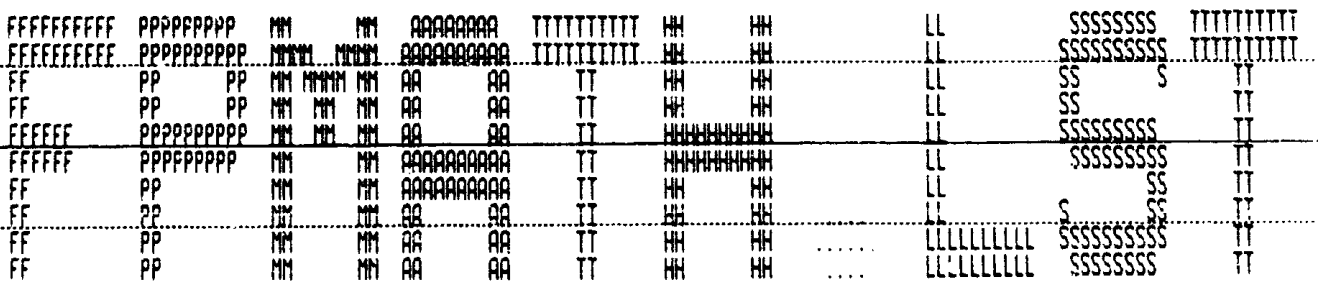


TABLE OF CONTENTS

```
                                .TITLE  FPMATH  FLOATING POINT ROUTINES
                                IDENT   /MEL032/
                                .SBTTL  FLOATING POINT ROUTINES
                                .GLOBL  ITSINT
                                .GLOBL  CNT.EQU.TFLGS1          ;MARKS STUFF
                                .GLOBL  ERRCD,X5,X4,X3,X2,X1,X0
                                .GLOBL  FUZA,FUZB,FUZC
                                .GLOBL  NGTAB,P'IPS,FUZZLE
                                .GLOBL  Y5,Y4,Y3,Y2,Y1,Y0,T1,T2,T3,SIGNS
                                .GLOBL  SIGNS,A0X,UNDER,NORM,ASCFPN,MULPNT
                                .GLOBL  DIGFLG,N12,NS,E1,E2,E3,ES,DREXIB
                                .GLOBL  A8X,A5X,A6X,A11X,VALTG,A7X,ZANS,MAXANS
                                .GLOBL  DEXP,GETCHR,CHAR,T4,TABPNT,SHNR
                                .GLOBL  PSHRET,RTRN,ZIX
                                .GLOBL  FPADD,FPSUB,FPMUL,FPDIV,FIX1,FLOAT1,FLOAT2
                                .GLOBL  A10X,A12X,A13X,EFLAG
                                .GLOBL  PSHFPN,PULFPN,FPNASC,RNDATA,CMPFPN
                                .GLOBL  ERFIXN  ;FIX NEG. NUMBER
                                .GLOBL  ERFXOV  ;FIX OVERFLOW
                                .GLOBL  ERFPOV  ;FP OVERFLOW
                                .GLOBL  ERFPDV  ;DIVIDE BY 0
0000'                           .CSECT  FPMATH
                        ;xxxxxxxxx FPSI xxxxxxxxxx
                                .SBTTL  FIX1A   PRODUCES THE INTERGER PART OF A FPN
                        ;FIX1 PRODUCES THE INTEGER PART OF A FLOATING
                        ;POINT NUMBER  THE INDEX REGISTER POINTS TO
                        ;THE FLAG IN THE BYTE IMIDIATLY PRECEEDING THE EXPONENT
                        ;OF THE FLOATING POINT NUMBER.  THE RESULT IS LEFT IN
                        ;TOP 2 BYTES OF THE MANTISSA, OR IX+3 AND IX+4. THE POSSIBLE
                        ;ERRORS ARE A NEGATIVE ARG, WHICH STILL PRODUCES THE
                        ; INT. PART OF THE ABS. VALUE, AND AN ARG >2 16-1.  THE
                        ; INDEX REGISTER IS LEFT UNCHANGED
                        ; AS OF VERSION 010, TAKES THE ROUNDED INSTEAD OF THE
                        ; TRUNCATED PART
                        ;
                        ;FIX1 RETURNS THROUGH A COMMON EXIT POINT WHICH DOES A "LDA A  ERRCD,0"
                        ;TO SET THE CONDITION CODES FOR THE CALLING ROUTINE.    - JDG
                        ;
0000    A6  01          FIX1:   LDA A   1,X     ;EXP HIGH
0002    2B  41                  BMI     NEGERR  ;NEGATIVE
0004    80  04          FIX1A:  SUB A   4,I
0006    2B  36                  BMI     ZFIX    ;EXP IS <0, SO RETURN A 0
0008    26  43                  BNE     OERR    ;EXP IS GROSSLY LARGE
000A    C6  08                  LDA B   8.I
000C    A6  03                  LDA A   3,X             ;NO MATTER WHAT, NEED UPPER BYTE
                                                        ;OF THE EXPONENT, SO SET TO A
000E    E0  02                  SUB B   2,X             ;8-EXP
0010    23  0D                  BLS     1$
0012    44              2$:     LSR A                   ;EXP 0-7, ACCB IS 8-EXP
                                                        ;ALL DATA IS IN THE BYTE IN ACCA
0013    5A                      DEC B
0014    26  FC                  BNE     2$
0016    24  01          3$:     BCC     4$              ;CHECK ROUNDING BIT
0018    4C                      INC A                   ;CANT OVERFLOW
0019    97  04          4$:     STA A   4,X
001B    E7  03                  STA B   3,X             ;B IS A 0, NOTE
001D    20  23                  BRA     FIXRTN          ;EXIT THRU COMMON EXIT POINT.
```

```
001F   C8  08   1$:      ADD B  8.,I           ; B IS 16-EXP
0021   27  17            BEQ    5$
0023   C1  09            CMP B  9.,I
0025   24  26            BCC    OERR           ;EXP > 16.
0027   44       6$:      LSR A
0028   66  04            ROR    4,X            ;TAKE CARE OF BOTH
002A   5A                DEC B
002B   26  FA            BNE    6$
002D   24  07   8$:      BCC    7$             ;ROUNDING BIT
002F   6C  04            INC    4,X
0031   26  03            BNE    7$
0033   4C                INC A
0034   27  17            BEQ    OERR           ;65K + A SMIDGEN(YETCH)
0036   A7  03   7$:      STA A  3,X
0038   20  08            BRA    FIXRTN         ;EXIT THRU COMMON EXIT POINT.

003A   69  05   5$:      ROL    5,X            ;GET ROUNDING BIT FOR EXP=16
003C   20  EF            BRA    8$
003E   6F  03   ZFIX:    CLR    3,X
0040   6F  04            CLR    4,X    ;ANSWER IS 0
0042   96  00G  FIXRTN:  LDA A  ERRCD,D        ;LOAD ACC-A WITH THE ERRCD TO SET THE
                                               ;CONDITION CODES.
0044   39       FIXRN1:  RTS                   ;BYE
0045   C6  00G  NEGERR:  LDA B  ERFIXN,I       ;FLAG ERROR
0047   D7  00G           STA B  ERRCD,D
0049   84  07            AND A  7,I    ;MAKE PLUS
004B   20  B7            BRA    FIX1A  ;FIX ABS VALUE
004D   C6  00G  OERR:    LDA B  ERFXOV,I
004F   D7  00G           STA B  ERRCD,D ;FLAG IT
0051   86  FF            LDA A  377,I
0053   A7  03            STA A  3,X
0055   A7  04            STA A  4,X    ;SET MAX INTEGER
0057   20  E9            BRA    FIXRTN         ;EXIT THRU COMMON EXIT POINT.
```

```
                                                .SBTTL  FLOAT1  PRODUCES A 9 BYTE FPN FROM A 3 BYTE INTERGER
                                ; xxxxxxxxxxxx FLOAT 1 xxxxxxxxxxxxxxxx
                                ;
                                ; THE FOLLOWING ROUTINE TAKES THE 3 BYTE INTERGER FROM THE STACK
                                ;     AND RETURNS A 9 BYTE FLOATING POINT NUMBER ON THE STACK.
                                ;
                                ;
0059                            FLOAT1:
0059    BD      0000G           FLOAT2: JSR     PSHRET
005C    32                              PUL A                   ;THROW AWAY TAG
005D    4F                              CLR A                   ;SET REST
005E    36                              PSH A
005F    36                              PSH A
0060    36                              PSH A
0061    36                              PSH A                   ;   TO ZERO
0062    C6      30                      LDA B   48.,I           ;SET EXPONENT
0064    37                              PSH B                   ;   TO 48.
0065    86      04                      LDA A   4,I
0067    36                              PSH A
0068    86      00G                     LDA A   VALTG,I         ;TAG IT.
006A    36                              PSH A
0068    7E      01A1'                   JMP     NORM

                                        .SBTTL  SHMR    SHIFT MANTISSA RIGHT

                                ;THE FOLLOWING ROUTINE, ENTERED AT SHMR  SHIFTS THE
                                ;48 BIT VALUE AT LOCS. X+3 TO X+8 RIGHT ACCH PLACES
006E    64      03      SHLP:   LSR     3,X
0070    66      04              ROR     4,X
0072    66      05              ROR     5,X
0074    66      06              ROR     6.,X
0076    66      07              ROR     7.,X
0078    66      08              ROR     8.,X
007A    4A                      DEC     A       ;ONE BIT SHIFT
007B    85      07      SHMR:   BIT A   7,I     ;NEED ONE BIT SHIFTS?
007D    26      EF              BNE     SHLP    ;YUP
007F    40                      TST A   ;NEED 8 BITTERS?
0080    27      C2      16$:    BEQ     FIXRN1          ;YES
0082    E6      07              LDA B   7,X
0084    E7      08              STA B   8.,X
0086    E6      06              LDA B   6.,X
0088    E7      07              STA B   7.,X
008A    E6      05              LDA B   5,X
008C    E7      06              STA B   6.,X
008E    E6      04              LDA B   4,X
0090    E7      05              STA B   5,X
0092    E6      03              LDA B   3,X
0094    E7      04              STA B   4,X
0096    6F      03              CLR     3,X
0098    80      08              SUB A   8.,I
009A    20      E4              BRA     16$
```

```
                                        .SBTTL  FPSUB   FLOATING POINT SUBTRACT
009C    30                      FPSUB:  TSX
009D    A6  03                  LDA A   3,X     ;FLIP SIGN OF B, AND DO ADD
009F    88  80                  EOR A   200,I
00A1    A7  03                  STA A   3,X
                                        .SBTTL  FPADD   FLOADING POINT ADD
00A3    BD  0000G       FPADD:  JSR     PSHRET
00A6    A6  01                  LDA A   1,X             ;SAVE SIGN/EXPOF B
00A8    97  00G                 STA A   X0,D
00AA    84  07                  AND A   7,I
00AC    A7  01                  STA A   1,X             ;MAKE ABS VALUE
00AE    A6  03                  LDA A   3,X             ;CHECK B=0
00B0    27  24                  BEQ     BANS1           ;YES
00B2    A6  0C                  LDA A   12.,X           ;IS A=0?
00B4    26  27                  BNE     ALIGN   ;NO
00B6    A6  08          BANS:   LDA A   8.,X    ;B IS THE ANSWER, COPY B TO A
00B8    A7  11                  STA A   17.,X
00BA    A6  07                  LDA A   7.,X
00BC    A7  10                  STA A   16.,X
00BE    A6  06                  LDA A   6,X
00C0    A7  0F                  STA A   15.,X
00C2    A6  05                  LDA A   5.,X
00C4    A7  0E                  STA A   14.,X
00C6    A6  04                  LDA A   4.,X
00C8    A7  0D                  STA A   13.,X
00CA    A6  03                  LDA A   3,X
00CC    A7  0C                  STA A   12.,X
00CE    96  00G                 LDA A   X0,D    ;EXP WAS SAVED IN X0 FOR THIS PURPOSE
00D0    A7  0A                  STA A   10.,X
00D2    A6  02                  LDA A   2,X
00D4    A7  08                  STA A   11.,X
00D6    BD  0000G       BANS1:  JSR     A9X     ;PRUNE B OFF STACK
00D9    35                      TXS
00DA    7E  0000G               JMP     RTRN    ;QUIT
00DD                    ALIGN:
                        ;ALIGN LINES THE TWO FPNS UP SO THE THEY HAVE
                        ;THE SAME EFFECTIVE EXPONENT.
                        ;
                        ; FINDS THE DIFFERENCE IN EXPONENTS, AND SHIFTS
                        ; THE ONE WITH THE SMALLEST EXPONENT RIGHT
                        ; THAT MANY BITS
00DD    A6  0B                  LDA A   11.,X   ;COMPUTE DIFF IN EXP
00DF    A0  02                  SUB A   2,X
00E1    E6  09                  LDA B   10.,X
00E3    C4  07                  AND B   7,I             ;GET JUST EXP
00E5    E2  01                  SBC B   1,X     ;INTO ACC'S B AND A
00E7    2A  19                  BPL     3$
00E9    5C                      INC B           ;B EXP LARGER THAN A'S
00EA    26  CA                  BNE     BANS    ;B>>A
00EC    81  01                  CMP A   -47.,I  ;IS DIFF >47
00EE    25  C6                  BCS     BANS    ;YUP
00F0    40                      NEG A
00F1    E6  0A                  LDA B   10.,X   ;EXP OF RESULT IS THAT OF
                                                ;LARGER NUMBER,OR THAT OF B
00F3    C4  80                  AND B   200,I
00F5    EA  01                  ORA A   1,X     ;SAVE SIGN ON A
00F7    E7  0A                  STA B   10.,X
```

```
00F9    E6   02                 LDA B   2,X
00FB    E7   0B                 STA B   11.,X
00FD    BD   0000G              JSR     A9X     ;MOVE X SO THAT IT POINTS AT OP A
0100    20   06                 BRA     4$
0102    26   02          3$:    BNE     BANS1   ;A))B
0104    81   30                 CMP A   48.,I
0106    24   CE                 BCC     BANS1
                         ; IF THE EXPONTNET OF A IS MUCH GREATER THAN
                         ; THAT OF B, OR IF B=0 , THEN THROW AWAY B AND RETURN
0108    BD   0078'       4$:    JSR     SHMR    ;SHIFT MANTISSA RIGHT ACCA PLACES
010B    30                      TSX             ;DO ADD
                         ;
                         ; IF SIGNS OF NUMBERS ARE DIFFERENT,USE
                         ; A SUBTRACT OF A FROM B AND FIGURE OUT WHO WAS
                         ; GPEATER IN ABS. VALUE AT A LATER DATE/
                         ;
                         ; IF THE SIGNS ARE THE SAME, JUST ADD AND BE
                         ; CARFULL ABOUT POSSIBLE OVERFLOW, FORCING A SHIFT BACK
                         ; TO THE RIGHT ONE BIT.
010C    96   00G                LDA A   X0.D
010E    A8   0A                 EOR A   10.,X
0110    2B   3F                 BMI     8$      ;SIGNS ARE DIFFERENT,SO DO SUB
0112    A6   08                 LDA A   8.,X    ;48 BIT ADD B MANT TO THE A MANT
0114    AB   11                 ADD A   17.,X
0116    A7   11                 STA A   17.,X
0118    A6   07                 LDA A   7,X
011A    A9   10                 ADC A   16.,X
011C    A7   10                 STA A   16.,X
011E    A6   06                 LDA A   6.,X
0120    A9   0F                 ADC A   15.,X
0122    A7   0F                 STA A   15.,X
0124    A6   05                 LDA A   5,X
0126    A9   0E                 ADC A   14.,X
0128    A7   0E                 STA A   14.,X
012A    A6   04                 LDA A   4.,X
012C    A9   0D                 ADC A   13.,X
012E    A7   0D                 STA A   13.,X
0130    A6   03                 LDA A   3.,X
0132    A9   0C                 ADC A   12.,X
0134    A7   0C                 STA A   12.,X
0136    24   12                 BCC     9$      ;NO OVERFLOW PROBLEMS IF NO CARRY

0138    66   0C                 ROR     12.,X   ;SHIFT MANT RIGHT ONE BIT
013A    66   0D                 ROR     13.,X
013C    66   0E                 ROR     14.,X
013E    66   0F                 ROR     15.,X
0140    66   10                 ROR     16.,X
0142    66   11                 ROR     17.,X
0144    6C   0B                 INC     11.,X   ;INC EXPONENT
0146    26   02                 BNE     9$
0148    6C   0A                 INC     10.,X
014A    BD   0000G       9$:    JSR     A9X
014D    35                      TXS             ;PRUNE STACK
014E    7E   01F1'              JMP     FRTST   ; EXIT GRACFULLY ,NO NEED TO NORMALIZE
0151    A6   11          8$:    LDA A   17.,X   ;SUB BMANT FROM A MANTISSA
0153    A0   08                 SUB A   8.,X
0155    A7   11                 STA A   17.,X
```

```
115  0157  A6  10          LDA A  16.,X
116  0159  A2  A2          SBC A  7.,X
117  015B  A7  10          STA A  16.,X
118  015D  A6  0F          LDA A  15.,X
119  015F  A2  06          SBC A  6.X
120  0161  A7  0F          STA A  15.,X
121  0163  A6  0E          LDA A  14.,X
122  0165  A2  05          SBC A  5.X
123  0167  A7  0E          STA A  14.,X
124  0169  A6  0D          LDA A  13.,X
125  016B  A2  04          SBC A  4..X
126  016D  A7  0D          STA A  13.,X
127  016F  A6  0C          LDA A  12.,X
128  0171  A2  03          SBC A  3..X
129  0173  A7  0C          STA A  12.,X
130                    ; IF CARRY IS SET, THE ABS(B)>ABS(A), AND
131                    ; WE MUST FLIP THE SIGN OF THE RESULTING
132                    ; SIGN BIT, AND NEGATE THE RESULT
133  0175  24  26          BCC    10$
134  0177  A6  0A          LDA A  10.,X
135  0179  88  80          EOR A  200,I
136  017B  A7  0A          STA A  10.,X   ;FLIP SIGN OF RESULT
137  017D  63  0C          COM    12.,X
138  017F  63  0D          COM    13.,X
139  0181  63  0E          COM    14.,X
140  0183  63  0F          COM    15.,X
141  0185  63  10          COM    16.,X
142  0187  60  11          NEG    17.,X
143  0189  26  12          BNE    10$     ;SURE IS A PAIN TO NEGATE, AINT IT
144  018B  6C  10          INC    16.,X
145  018D  26  0E          BNE    10$
146  018F  6C  0F          INC    15.,X
147  0191  26  0A          BNE    10$
148  0193  6C  0E          INC    14.,X
149  0195  26  06          BNE    10$
150  0197  6C  0D          INC    13.,X
151  0199  26  02          BNE    10$
152  019B  6C  0C          INC    12.,X
153  019D  BD  0000G   10$: JSR    A5X
154  01A0  35              TXS            ;PRUNE STACK
```

```
                                        .SBTTL  NORM    NORMALIZE FPN
                                        .GLOBL  ZIXM
                                ;NORMALIZE THE FLOATING POINT NUMBER ON
                                ;THE TOP OF THE STACK.  THE RETURN ADREESS FOR NORM IS
                                ;NOT ON THE STACK, BUT IS IN ENRET.  THE ROUTINE WILL
                                ;TRY TO DO 8 BIT SHIFTS IF POSSIBLE, BY MOVING WHOLE
                                ;BYTES, AND WILL THEN RESORT TO 1 BIT RIPPLE SHIFTS
                                ;UNTIL THE TOP BIT IS A ONE.  THE EXPONENT IS THEN
                                ; UPDATED TO NOTE THE SHIFTS, AND THE EXPONENT BYTE
                                ; ARE TESTED FOR OVER/UNDER FLOW.  THE SIGN BIT IS INSERTED
                                ; FROM THE TOP BIT OF SIGMS.
01A1  5F              NORM:     CLR B           ;NORMALIZE-B=SHIFT COUNTER
01A2  30                        TSX
01A3  6D  03          NORM1:    TST     3,X     ;IS THERE ANY HOPE THAT WERE DONE?
01A5  26  3A                    BNE     NORM2   ;YUP, TOP BYTE OF MANTISSA IS <>0
01A7  CB  08                    ADD B   8.,I    ;NO, SHIFT LEFT 8 BITS
01A9  C1  30                    CMP B   48.,I   ;IF 48 BITS ARE 0, THEN
01AB  26  0F                    BNE NORM3
01AD  6F  01                    CLR     1,X     ;A  IS ZERO, AND THE
01AF  6F  02                    CLR     2,X     ;ANSWER IS ALL 0
01B1  73  0000G                 COM     FUZR    ;NOTE PERFECT COMPARE FOR FUZZ
01B4  7E  0000G                 JMP     RTRN    ;PAINLESS, HUH
                                .GLOBL  NORMR
01B7  BD  0000G       NORMR:    JSR     PSHRET  ;FUNNY ENTRY TO NORM
01BA  20  E5                    BRA     NORM
01BC  A6  04          NORM3:    LDA A   4,X     ;SHUFFELBOARD ALL 6 BYTES UP ONE
01BE  A7  03                    STA A   3,X
01C0  A6  05                    LDA A   5,X
01C2  A7  04                    STA A   4,X
01C4  A6  06                    LDA A   6,X
01C6  A7  05                    STA A   5,X
01C8  A6  07                    LDA A   7,X
01CA  A7  06                    STA A   6,X
01CC  A6  08                    LDA A   8.,X
01CE  A7  07                    STA A   7,X
01D0  6F  08                    CLR     8.,X
01D2  20  CF                    BRA     NORM1
01D4  5C              NORM4:    INC B
01D5  68  08                    ASL     8.,X    ;ONE BIT SHIFTS
01D7  69  07                    ROL     7,X
01D9  69  06                    ROL     6,X
01DB  69  05                    ROL     5,X
01DD  69  04                    ROL     4,X
01DF  69  03                    ROL     3,X
01E1  2A  F1          NORM2:    BPL     NORM4   ;BRANCH IF NOT NORM. YET
01E3  F7  0000G                 STA B   FUZR    ;NOTE RANGE OF COMPARISON FOR FUZZ
01E6  A6  02                    LDA A   2,X     ;NORMAL, SUB ACCB FROM EXP.
01E8  10                        SBA
01E9  A7  02                    STA A   2,X
01EB  A6  01                    LDA A   1,X
01ED  82  00                    SBC A   0,I
01EF  A7  01                    STA A   1,X
01F1  A6  01          FRIST:    LDA A   1,X             ;TEST FOR OVER/UNDERFLOW
01F3  85  78                    BIT A   170,I
01F5  27  26                    BEQ     FPRET           ;OK, NONE OF THE SPARE BITS SET
01F7  85  40                    BIT A   100,I           ;DISCRIMINATE BETWEEN UNDER AND OVER
01F9  26  1F                    BNE     UNDER
```

```
                                        ;NOTE THAT IN THE LAST, THE SIGN OF THE NUMBER WAS CLOBBERED
                                        ;BUT IT REALLY DOSN'T MAKE ANY DIFFERENCE
01F8  86  000G    OVER:   LDA A   ERFPOV,I
01FD  97  000G            STA A   ERRCD,D         ;SET ERROR
01FF  30          MAXONS: TSX
0200  A6  01              LDA A   1,X
0202  84  80              AND A   200,I   ;SAVE SIGN
0204  8A  07              ORA A   7,I
0206  A7  01              STA A   1,X     ;SET MAX EXP
0208  86  FF              LDA A   377,I
020A  A7  02              STA A   2,X
020C  A7  03              STA A   3,X
020E  A7  04              STA A   4,X
0210  A7  05              STA A   5,X
0212  A7  06              STA A   6,X
0214  A7  07              STA A   7,X
0216  A7  08              STA A   8.,X
0218  20  03              BRA     FPRET   ;LARGEST NUMBER
021A                      UNDER:
021A  30          ZONS:   TSX
021B  8D  03              BSR     ZIX
021D  7E  0000G   FPRET:  JMP     RTRN
0220  6F  05      ZIX:    CLR     5,X
0222  6F  06              CLR     6,X
0224  6F  07              CLR     7,X
0226  6F  08              CLR     8.,X
0228  6F  01      ZIX4:   CLR     1,X     ;SHIF A 0
022A  6F  02              CLR     2,X
022C  6F  03              CLR     3,X
022E  6F  04              CLR     4,X
0230  39                  RTS
```

```
                                        .SBTTL  FPDIV   FLOATING POINT DIVIDE
                                ;
                                ; THIS DIVIDE IS NON- RESTORING.
                                ; THE ALGORTHIM STARTS WITH THE DIVIDEND IN X, AND
                                ; THE DIVISOR IN Y.
                                ; FOR THE FIRST BIT, Y IS SUBTRACTED FROM X.
                                ; FROM THERE ON, Y IS SUBTRACTED IF THE LAST BIT
                                ; OF THE QUOTIENT IS 1, AND ADDED IF THE LAST BIT
                                ; WAS A ZERO. REGISTER B IS USED TEMPORARILY AS A WORK
                                ; REGISTER, AND THE TOP BIT IS USED TO GENERATE THE
                                ; NEXT BIT OF THE QUOTIENT. IT ALSO STORES THE OLD BIT
                                ; LAST SHIFTED INTO THE QUOTIENT SO THAT THE ADD/SUB
                                ; DECISION MAY BE MADE. ACCA IS THE LOOP COUNTER AS
                                ; WELL AS THE TEMPORY STORAGE FOR BITS OF THE QUOTIENT
                                ; UNTIL 8 OF THEM HAVE BEEN PRODUCED.  X REGISTER IS
                                ; A POINTER INTO THE STACK SHOWING WHERE THE NEXT BYTE OF THE
                                ; QUOTIENT IS TO BE PLACED.  T2 IS A COUNTER, & LONG, TO
                                ; DETERMINE WHEN THE DIV. IS DONE.  T1 CONTAINS THE
                                ; BIT SHIFTED OUT OF THE DIVIDEND IN ITS UPPER POSITION,
                                ; AND AIDS IN DETERMINING THE NEXT BIT OF THE QUOTIENT.
                                ; THIS ROUTINE WAS WRITTEN FOR SPEED, NOT EASE OF UNDERSTANDING.
                                ;
0231  BD  0000G     FPDIV:  JSR     PSHRET
0234  A6  03                LDA A   3,X             ;CHECK FOR DIVIDE BY 0
0236  26  0A                BNE     1$
0238  BD  0000G             JSR     A9%
0238  35                    TXS
023C  86  00G               LDA A   ERFPDV,I
023E  97  00G               STA A   ERRCD.D         ;SET ERROR
0240  20  BD                BRA     MAXANS          ;AND RETURN +-INF
0242  A6  0C        1$:     LDA A   12.,X   ;CHECK FOR 0/X=0
0244  26  06                BNE     2$
0246  BD  0000G             JSR     A9%
0249  35                    TXS
024A  20  D1                BRA     FPRET   ;YES, JUST THROW AWAY X
024C  A6  08        2$:     LDA A   11.,X
024E  A0  02                SUB A   2,X     ;EXP IS DIFFERENCE OF EXPONENTS
0250  E6  0A                LDA B   10.,X
0252  E2  01                SBC B   1,X     ;AND SIGN IS EOR OF SIGNS
0254  8B  01                ADD A   1,I     ;PLUS ONE, AS DIVIDE IS OFF BY FACTOR OF 2
0256  C9  04                ADC B   4,I     ;AND RESTORE THE BIAS BIT
0258  A7  08                STA A   11.,X
025A  E7  0A                STA B   10.,X
025C  32                    PUL A
025D  32                    PUL A
025E  32                    PUL A
025F  32                    PUL A           ;MOVE MANT OF DIVISOR TO Y5-Y0
0260  97  00G               STA A   Y5,D
0262  32                    PUL A
0263  97  00G               STA A   Y4,D
0265  32                    PUL A
0266  97  00G               STA A   Y3,D
0268  32                    PUL A
0269  97  00G               STA A   Y2,D
026B  32                    PUL A
026C  97  00G               STA A   Y1,D
026E  32                    PUL A
```

```
026F  97  00G           STA A  Y0.D
0271  30                TSX
                        ;GET DIVIDEND INTO X5-X0
0272  A6  03            LDA A  3,X
0274  97  00G           STA A  X5.D
0276  A6  04            LDA A  4,X
0278  97  00G           STA A  X4.D
027A  A6  05            LDA A  5,X
027C  97  00G           STA A  X3.D
027E  A6  06            LDA A  6,X
0280  97  00G           STA A  X2.D
0282  A6  07            LDA A  7,X
0284  97  00G           STA A  X1.D
0286  A6  08            LDA A  8,X
0288  97  00G           STA A  X0.D
                        ; PERFORM DIVIDE IN PAGE 0, PUTTING QUOTIENT BACK ONTO
                        ; STACK AS GENERATED
028A  86  06      MDV:  LDA A  6,I   ;NO. OF BYTES IN QOUOTIENT
028C  97  00G           STA A  T2.D
028E  97  00G           STA A  T1.D  ;BIT 7 OF T1 CLEARED, WHICH IS CARRY OUT BIT FOR SHIFTS
0290  86  01            LDA A  1,I   ;1 BIT IS LOOP COUNTER
0292  D6  00G     1$:   LDA B  X0.D  ;TRY A SUB
0294  D0  00G           SUB B  Y0.D
0296  D7  00G           STA B  X0.D
0298  D6  00G           LDA B  X1.D
029A  D2  00G           SBC B  Y1.D
029C  D7  00G           STA B  X1.D
029E  D6  00G           LDA B  X2.D
02A0  D2  00G           SBC B  Y2.D
02A2  D7  00G           STA B  X2.D
02A4  D6  00G           LDA B  X3.D
02A6  D2  00G           SBC B  Y3.D
02A8  D7  00G           STA B  X3.D
02AA  D6  00G           LDA B  X4.D
02AC  D2  00G           SBC B  Y4.D
02AE  D7  00G           STA B  X4.D
02B0  D6  00G           LDA B  X5.D
02B2  D2  00G           SBC B  Y5.D
02B4  D7  00G           STA B  X5.D
02B6  56                ROR B
02B7  53                COM B        ;BIT 7 SET IF SUB WENT WITHOUT UNDERFLOW
02B8  DA  00G           ORA B  T1.D  ;IF THAT OR SHIFT OUT BIT WAS 1, THEN Q BIT IS 1
02BA  59          2$:   ROL B
02BB  49                ROL A        ;SHIFT BIT INTO DIVISOR
02BC  16                TAB          ;REMEMBER Q BIT
02BD  24  00            BCC    3$    ;WAIT FOR BIT TO COME OUT TOP
02BF  A7  03            STA A  3,X   ;8 BITS OF Q
02C1  08                INX
02C2  86  01            LDA A  1,I   ;START LOOP COUNTER AGAIN
02C4  7A  0000G         DEC    T2
02C7  26  03            BNE    3$
02C9  7E  01A1'         JMP    NORM  ;DONE
02CC  78  0000G   3$:   ASL    X0    ;SHIFT DIVIDEND
02CF  79  0000G         ROL    X1
02D2  79  0000G         ROL    X2
02D5  79  0000G         ROL    X3
02D8  79  0000G         ROL    X4
```

```
115  02D8  79  0000G          ROL   X5
116  02DE  56                  ROR   B
117  02DF  D7  00G             STA B T1,D    ;SAVE SHIFT OUT BIT
118  02E1  25  AF              BCS   1$      ;IF LAST BIT OF Q WAS A 1,
119                                          ;DO ANOTHER SUB. ELSE TRY A ADD
120  02E3  D6  00G             LDA B X0,D    ;DO ADD
121  02E5  D8  00G             ADD B Y0,D
122  02E7  D7  00G             STA B X0,D
123  02E9  D6  00G             LDA B X1,D
124  02EB  D9  00G             ADC B Y1,D
125  02ED  D7  00G             STA B X1,D
126  02EF  D6  00G             LDA B X2,D
127  02F1  D9  00G             ADC B Y2,D
128  02F3  D7  00G             STA B X2,D
129  02F5  D6  00G             LDA B X3,D
130  02F7  D9  00G             ADC B Y3,D
131  02F9  D7  00G             STA B X3,D
132  02FB  D6  00G             LDA B X4,D
133  02FD  D9  00G             ADC B Y4,D
134  02FF  D7  00G             STA B X4,D
135  0301  D6  00G             LDA B X5,D
136  0303  D9  00G             ADC B Y5,D
137  0305  D7  00G             STA B X5,D
138                            ;NEXT Q BIT IS 1 IF ADD OVERFLOWS AND THE LAST
139                            ;SHIFT PRODUCED A 1
140  0307  56                  ROR B
141  0308  D4  00G             AND B T1,D
142  030A  20  AE              BRA   2$
```

```
                                        .SBTTL MULTPLY  FLOATING POINT


                                ;X0:    .BYTE ,,,,,     ;MULTIHEND
                                ;Y0:    .BYTE ,,,,,     ;RESULT
                                ;T1:    .BYTE           ;COUNTER FOR # OF MULTPLIER BYTES
                                ;T2:    .BYTE           ;=0 IF MULTIPLIER BITS 0 SO FAR

030C    CE  0000        FPMUL:  LDX     0,I
030F    DF    00G               STX     Y5,D    ;CLEAR RESULT TO 0
0311    DF    00G               STX     Y3,D
0313    DF    00G               STX     Y1,D
0315    DF    00G               STX     T1,D
0317    BD  0000G               JSR     PSHRET
031A    A6    0C                LDA A   12.,X
031C    26    07                BNE     1$              ;CHECK FOR 0*X,X*0
031E    BD  0000G       2$:     JSR     A9X
0321    35                      TXS
0322    7E  021A'               JMP     ZANS
0325    A6    03        1$:     LDA A   3,X
0327    27    F5                BEQ     2$
                        ;NIETHER IS 0
0329    32                      PUL A   ;TAG
032A    33                      PUL B
032B    32                      PUL A   ;EXP
032C    AB    08                ADD A   11.,X
032E    E9    0A                ADC B   10.,X   ;EXP IS SUM OF EXPS
0330    C0    04                SUB B   4,I     ;TAKE CARE OF BIAS BITS
0332    A7    0B                STA A   11.,X
0334    E7    0A                STA B   10.,X
0336    32                      PUL A           ;STORE MANTISA
0337    97    00G               STA A   X5,D
0339    32                      PUL A
033A    97    00G               STA A   X4,D
033C    32                      PUL A
033D    97    00G               STA A   X3,D
033F    32                      PUL A
0340    97    00G               STA A   X2,D
0342    32                      PUL A
0343    97    00G               STA A   X1,D
0345    32                      PUL A
0346    97    00G               STA A   X0,D
0348    86    06                LDA A   6,I     ;INIT COUNTERS & FLAG
034A    97    00G               STA A   T1,D
034C    E6    11        AGAIN:  LDA B   17.,X
034E    26    1C                BNE     10$

0350    D1    00G               CMP B   T2,D    ;MULTIPLIER BYTE=0
0352    27    57                BEQ     NEXTBY
0354    96    00G               LDA A   Y1,D            ;ROTATE RESULT ONE BYTE RIGHT
0356    97    00G               STA A   Y0,D
0358    96    00G               LDA A   Y2,D
035A    97    00G               STA A   Y1,D
035C    96    00G               LDA A   Y3,D
```

```
    58  035E  97  00G             STA A   Y2,D
    59  0360  96  00G             LDA A   Y4,D
    60  0362  97  00G             STA A   Y3,D
    61  0364  96  00G             LDA A   Y5,D
    62  0366  97  00G             STA A   Y4,D
    63  0368  D7  00G             STA B   Y5,D
    64  036A  20  3F              BRA     NEXTBY
    65
    66  036C  D7  00G     10$:    STA B   T2,D    ;SET NON ZERO BYTE SEEN
    67  036E  0D                  SEC             ;MULTIPLY BY MULTIPLIER BYTE
    68                                            ;B REG IS ALSO THE LOOP COUNTER
    69                                            ;WHEN THE BIT FROM THE CARRY ABOVE
    70                                            ;SHIFTS THROUGH LEAVING 0, THE BYTE IS FINISHED
    71  036F  56                  ROR B
    72  0370  24  24      NEXT:   BCC     MURETU
    73  0372  96  00G     ADD:    LDA A   Y0,D    ;ADD 4 BYTES TO THE RESULT
    74  0374  9B  00G             ADD A   X0,D
    75  0376  97  00G             STA A   Y0,D
    76  0378  96  00G             LDA A   Y1,D
    77  037A  99  00G             ADC A   X1,D
    78  037C  97  00G             STA A   Y1,D
    79  037E  96  00G             LDA A   Y2,D
    80  0380  99  00G             ADC A   X2,D
    81  0382  97  00G             STA A   Y2,D
    82  0384  96  00G             LDA A   Y3,D
    83  0386  99  00G             ADC A   X3,D
    84  0388  97  00G             STA A   Y3,D
    85  038A  96  00G             LDA A   Y4,D
    86  038C  99  00G             ADC A   X4,D
    87  038E  97  00G             STA A   Y4,D
    88  0390  96  00G             LDA A   Y5,D
    89  0392  99  00G             ADC A   X5,D
    90  0394  97  00G             STA A   Y5,D
    91  0396  76  0000G   MURETU: ROR     Y5              ;ROTATE RESULT 1 BIT RIGHT
    92  0399  76  0000G           ROR     Y4
    93  039C  76  0000G           ROR     Y3
    94  039F  76  0000G           ROR     Y2
    95  03A2  76  0000G           ROR     Y1
    96  03A5  76  0000G           ROR     Y0
    97
    98  03A8  54                  LSR B           ;GET NEXT BIT,CHECK IF DONE
    99  03A9  26  C5              BNE     NEXT
   100  03AB  09          NEXTBY: DEX
   101  03AC  7A  0000G           DEC     T1
   102  03AF  26  9B              BNE     AGAIN
   103                                            ;FINISHED MULTIPLING MANTISA
   104
   105
   106  03B1  30                  TSX             ;RETURN INDEX TO EXPONENT
   107
   108  03B2  96  00G     RETURN: LDA A   Y5,D    ;RETURN RESULT TO THE STACK
]]]]]
   109  03B4  A7  03              STA A   3,X
   110  03B6  96  00G             LDA A   Y4,D
   111  03B8  A7  04              STA A   4,X
   112  03BA  96  00G             LDA A   Y3,D
   113  03BC  A7  05              STA A   5,X
   114  03BE  96  00G             LDA A   Y2,D
```

```
 115  03C0  A7  06      STA A  6,X
 116  03C2  96  00G     LDA A  Y1+0
 117  03C4  A7  07      STA A  7,X
 118  03C6  96  00G     LDA A  Y0+0
 119  03C8  A7  08      STA A  8,X
 120  03CA  7E  01A1'   JMP    NORM
 121
 122
```

```
                                        .SBTTL  ASCFPN    CONVERT ASCII STRING TO FPN
                                        ;
                                        ;
                                        ;       T1 - DOT SEEN
                                        ;       T2,T3-OFFSET TO EXPONENT
                                        ;       T4- NOTES OVER/UNDER FLOWS
                                        ;       SIGNS-KEEP TRACK OF SIGN OF NUMBER,EXPONENT
                                        ;
03CD    BD      0000G   ASCFPN: JSR     PS%RET
                                        ;
                                        ;SET UP FLAGS FOR LEX
                                        ;IF CNT=1 THEN
                                        ;IF NUMBER IS AN INTEGER THEN
                                        ;CLEAR BIT 7 AND 2 OF TFLGS1 ELSE
                                        ;SET EQU=1
                                        ; END END
03D0    C6      7B                      LDA B   173,I           ;MASK FOR TFLAGS
03D2    F7      0000G                   STA B   DIGFLG          ;SET NO DIGITS SEEN FLAG
03D5    F7      0000G                   STA B   ITSINT          ;SET IS AN INTEGER UNTIL KNOW OTHERWOSE
03D8    96      00G                     LDA A   CNT,D
03DA    4A                              DEC A
03DB    26      04                      BNE     1$
03DD    D4      00G                     AND B   TFLGS1,D        ;CNT=1,ASSUME INTEGER UNLESS
                                                                ;OTHERWISE KNOWN
03DF    D7      00G                     STA B   TFLGS1,D
03E1    CE      0000    1$:     LDX     0,I
03E4    DF      00G                     STX     X5,D            ;CLEAR
03E6    DF      00G                     STX     X3,D            ;       ALL
03E8    DF      00G                     STX     X1,D            ;       APPROPIATE
03EA    DF      00G                     STX     T1,D
03EC    DF      00G                     STX     T3,D
03EE    DF      00G                     STX     DEXP,D
03F0    7F      0000G                   CLR     EFLAG           ;
03F3    7F      0000G                   CLR     SIGNS           ;       STUFF.
03F6    96      00G                     LDA A   CHAR,D          ;GET FIRST CHARACTER.
03F8    81      2B                      CMP A   '+,I
03FA    27      0B                      BEQ     MANTLP          ;IGNORE IF "+".
03FC    81      2D                      CMP A   '-,I
03FE    26      0A                      BNE     MANTL1
0400    BD      0537'                   JSR     NOTINT
0403    0D                              SEC
0404    76      0000G                   ROR     SIGNS           ;SET "-" SEEN IN NUMBER
0407    BD      0000G   MANTLP:  JSR     GETCHR          ;GET NEXT CHARACTER.
040A    80      30      MANTL1:  SUB A  '0,I            ;MAKE INTO BINARY
040C    24      0F                      BCC     1$
040E    81      FE                      CMP A   '.-'0,I         ;NOT A NUMBER AFTER ALL
0410    26      78                      BNE     TWO             ;WAS IT A "."?
0412    D6      00G                     LDA B   T1,D            ;YES. THE FIRST ONE?
0414    26      77                      BNE     TWO             ;NO, LET'S QUIT.
0416    BD      0537'                   JSR     NOTINT          ;TELL LEX IT WASN'T AN INTEGER
0419    97      00G                     STA A   T1,D            ;SET T1<>0
041B    20      EA                      BRA     MANTLP          ;GO AGAIN.


041D    81      0A      1$:     CMP A   10.,I           ;MAKE SURE IT'S A NUMBER.
```

```
    58  041F  24  1C            BCC   TRYE          ;IF NO, SET'S TRY AN "E".
    59  0421  7F  0000G         CLR   DIGFLG        ;FLAG A DIGIT SEEN
    60  0424  D6  00G           LDA B T1,D          ;BEFORE OR AFTER "."
    61  0426  26  05            BNE   2$
    62  0428  DE  00G           LDX   T2,D
    63  042A  08                INX
    64  042B  DF  00G           STX   T2,D          ;FIX UP EXPONENT
    65  042D  D6  00G       2$: LDA B X5,D          ;WILL
    66  042F  C1  15            CMP B 25,I          ;      IT GO?
    67  0431  24  D4            BCC   MANTLP        ;NO, BUT CONTINUE SCAN.
    68  0433  BD  058F'          JSR   TMULT         ;YES. MULTIPLY OLD BY 10.
    69                                               ;AND ADD IN NEW IN A
    70  0436  DE  00G           LDX   T2,D
    71  0438  09                DEX
    72  0439  DF  00G           STX   T2,D          ;NOTE IN EXPONENT
    73  043B  20  CA            BRA   MANTLP        ; SPLIT
    74
    75
    76
    77  043D  81  35        TRYE: CMP A  H65-'0,I
    78  043F  27  04            BEQ   1$            ;TEST FOR LOWCASE E
    79  0441  81  15            CMP A 'E-'0,I       ;WAS IT AN "E"?
    80  0443  26  48            BNE   TWO           ;NOPE
    81  0445  73  0000G     1$: COM   EFLAG         ;SET EFLAG IF ONLY AN E SEEN
    82                                               ;LEX CAN BACK UP IF SO AND
    83                                               ;EAT IT ITSELF
    84  0448  BD  0000G         JSR   GETCHR        ;SEE WHAT FOLLOWS AS NEXT CHARACTER.
    85  0448  81  2B            CMP A '+,I          ;IGNORE
    86  044D  27  0F            BEQ   2$            ;    "+" SIGN
    87  044F  81  20            CMP A 40,I          ;IGNORE
    88  0451  27  07            BEQ   2$            ;    A SPACE ALSO.
    89  0453  81  2D            CMP A '-,I          ;SEE IF IT WAS A MINUS
    90  0455  26  0C            BNE   4$            ;MUST BE SOMETHING ELSE.
    91  0457  7C  0000G         INC   SIGNS         ;NOTE EXPONENT NEGATIVE
    92  045A  BD  0537'      2$: JSR   NOTINT        ;NOTE THAT IT WASN'T AN INTEGR
    93  045D  7F  0000G         CLR   EFLAG         ;ALSO THAT A SOLITARY E WAS NOT SEEN
    94  0460  BD  0000G         JSR   GETCHR        ;GET ANOTHER ONE---
    95  0463  80  30        4$: SUB A '0,I
    96  0465  25  26            BCS   TWO           ;BRANCH IF NOT A NUMBER.
    97  0467  81  0A            CMP A 10.,I
    98  0469  24  22            BCC   TWO           ;BRANCH IF NOT A NUMBER.
]]]]]
    99  046B  D6  00G           LDA B DEXP,D        ;TOO BIG?
   100  046D  27  03            BEQ   5$
   101  046F  7C  0003G         INC   T4            ;FLAG IT.
   102  0472  D6  01G       5$: LDA B DEXP+1,D
   103  0474  C1  34            CMP B 52.,I
   104  0476  25  03            BCS   6$
   105  0478  7C  0000G         INC   T4            ;TOO BIG.
   106  047B  58            6$: ASL B
   107  047C  58                ASL B
   108  047D  DB  01G           ADD B DEXP+1,D      ;TIMES FIVE  (*5)
   109  047F  58                ASL B               ;TIMES TEN  (*10)
   110  0480  79  0000G         ROL   DEXP
   111  0483  1B                ABA
   112  0484  97  01G           STA A DEXP+1,D      ;  *10 + NEW
   113  0486  24  02            BCC   2$            ;GET NEXT
   114  0488  7C  0000G         INC   DEXP
```

```
        048B    20  CD              BRA    2$           ;SPLIT...............


        048D    96  00G     TWO:    LDA A  X5,D
        048F    9A  00G             ORA A  X4,D
        0491    9A  00G             ORA A  X3,D
        0493    9A  00G             ORA A  X2,D
        0495    27  03              BEQ    5$           ;FITS IN 16 BITS
        0497    BD  0537'           JSR    NOTINT       ;OTHERWISE, NOT AN INTERGER
        049A    D6  00G     5$:     LDA B  T2,D         ;MAKE DEXP AS T2,3+-DEXP
        049C    96  00G             LDA A  SIGNS,D      ;FIND OUT IF + OR -
        049E    46                  ROR A            ;THIS ONE SETS THE CARRY BIT FOR THE BCC INSTRUCTION.
        049F    96  00G             LDA A  T3,D         ;+, SO ADD
        04A1    24  06              BCC    2$           ;+
        04A3    90  01G             SUB A  DEXP+1,D
        04A5    D2  00G             SBC B  DEXP,D
        04A7    20  04              BRA    1$
        04A9    9B  01G     2$:     ADD A  DEXP+1,D
        04AB    D9  00G             ADC B  DEXP,D
        04AD    D7  00G     1$:     STA B  T3,D         ;SAVE REAL SIGN IN CASE
                                                        ;OF OVER/UNDER PROBLMS
        04AF    2A  08              BPL    3$           ;WAS POS
        04B1    70  0000G           NEG    T4           ;CHANGE OVER TO UNDERFLOWS
        04B4    53                  COM B
        04B5    40                  NEG A               ;TAKE ABS VALUE
        04B6    26  01              BNE    3$
        04B8    4C                  INC A
        04B9    C1  02      3$:     CMP B  2,I          ;TEST FOR EXPONENT TO BIG.
        04BB    25  04              BCS    4$
        04BD    D6  00G             LDA B  T3,D         ;TOO BIG.
        04BF    D7  00G             STA B  T4,D         ;NOTE IF OVER OR UNDER
        04C1    97  01G     4$:     STA A  DEXP+1,D     ;SAVE THE
        04C3    D7  00G             STA B  DEXP,D       ;    EXPONENT.
        04C5    CE  FFFEG           LDX    X5-2,I       ;MAKE THE MANTISSA FPN.
        04C8    BD  0549'           JSR    PSHFPN       ;STACK IT....
        04CB    30                  TSX
        04CC    86  30              LDA A  48.,I
        04CE    A7  02              STA A  2,X          ;SET EXPONENT TO "48".
        04D0    96  00G             LDA A  SIGNS,D      ;AND INSERT SIGN
        04D2    84  80              AND A  200,I
        04D4    8A  04              ORA A  4,I
        04D6    A7  01              STA A  1,X
        04D8    BD  01B7'           JSR    NORMR
        04DB    D6  00G     RETS:   LDA B  T4,D         ;OVER;UNDER PROBLEM??
        04DD    27  08              BEQ    1$           ;ALL IS OK
        04DF    2B  03              BMI    2$
        04E1    7E  01FE'           JMP    MAXANS
        04E4    7E  021A'   2$:     JMP    ZANS         ;UNDER
        04E7    86  06      1$:     LDA A  6,I          ;PROCEED TO MULTIPLY
        04E9    97  00G             STA A  T4,D         ;        6 CONSTANTS;
                                                        ;        10 +-8 < 10 +-256
        04EB    CE  0830'           LDX    PLTAB,I
        04EE    DF  00G             STX    TABPNT,D
        04F0    76  0000G           ROR    DEXP
        04F3    20  03              BRA    MLP1

```

```
  172  04F5  78  0001G  MLP:   ASL   DEXP+1
  173  04F8  24  0E     MLP1:  BCC   1$
  174  04FA  BD  0549'         JSR   PSHFP:.
  175  04FD  96  00G            LDA A T3.D
  176  04FF  2A  05             BPL   9$
  177  0501  BD  0231'         JSR   FPDIV
  178  0504  20  03            BRA   1$
  179  0506  BD  030C'  9$:    JSR   FPMUL
  180  0509  DE  00G    1$:    LDX   TABPNT.D
  181  050B  BD  0000G         JSR   ABX
  182  050E  DF  00G            STX   TABPNT.D
  183  0510  7A  0000G         DEC   T4
  184  0513  26  E0            BNE   MLP
  185  0515  96  01G           LDA A DEXP+1.D        ;3 BITS LEFT
  186  0517  27  13            BEQ   ASCDN           ;DONE WITH THIS
  187  0519  8B  20    LPABC:  ADD A 40.;            ;WHICH OF 10 7, 10 6,...10 4
  188  051B  26  15            BNE   ASCD1
  189  051D  BD  0549'         JSR   PSHFPN
  190  0520  96  00G           LDA A T3.D
  191  0522  2A  05            BPL   9$
  192  0524  BD  0231'         JSR   FPDIV
  193  0527  20  03            BRA   ASCDN
  194  0529  BD  030C'  9$:    JSR   FPMUL
  195  052C  7F  0000G  ASCDN: CLR   ERRCD           ;RETURN
  196  052F  7E  0000G         JMP   RTRN
  197  0532  BD  0000G  ASCD1: JSR   ABX
  198  0535  20  E2            BRA   LPABC
  199                    ;IF THE NO IS NOT AN INTEGER, DO THE FOLLOWING
  200  0537  7F  0000G  NOTINT: CLR   ITSINT
  201  053A  C6  01            LDA B 1,I
  202  053C  D1  00G           CMP B CNT.D           ;DO NOTHING IF CNT<>1
  203  053E  26  08            BNE   1$
  204  0540  D7  00G           STA B EQU.D
  205  0542  D6  00G           LDA B TFLGS1.D
  206  0544  CA  84            ORA B 204.I           ;SET FLAG RESET AT BEGINING
  207  0546  D7  00G           STA B TFLGS1.D
  208  0548  39        1$:    RTS
```

```
                                                .SBTTL  PSHFPN     PUSH A FLOADING POINT NUMBER
                0549    32                      PSHFPN: PUL A
                054A    97      01G             STA A   DREXTB+1,D
                054C    32                      PUL A
                054D    97      02G             STA A   DREXTB+2,D
                054F    A6      07              LDA A   7,X
                0551    36                      PSH A
                0552    A6      06              LDA A   6,X
                0554    36                      PSH A
                0555    A6      05              LDA A   5,X
                0557    36                      PSH A
                0558    A6      04              LDA A   4,X
                055A    36                      PSH A
                055B    A6      03              LDA A   3,X
                055D    36                      PSH A
                055E    A6      02              LDA A   2,X
                0560    36                      PSH A
                0561    A6      01              LDA A   1,X
                0563    36                      PSH A
                0564    A6      00              LDA A   0,X
                0566    36                      PSH A
                0567    86      00G             LDA A   VALTG,I
                0569    36                      PSH A
                056A    7E      0000G           JMP     DREXTB
```

```
                                .SBTTL  PULFPN    PULL FLOATING NUMBER
                                ;
                                ; THIS ROUTINE MOVES A FLOATING POINT NUMBER FROM THE
                                ;    STACK AND PUTS IT INTO THE MEMORY LOCATION
                                ;    POINTED TO BY THE X REGISTER.
                                ;
                                ;
0560  32            PULFPN: PUL A
056E  97  01G               STA A   DREXTB+1,O
0570  32                    PUL A
0571  97  02G               STA A   DREXTB+2,O
0573  31                    INS                     ;THROW AWAY TAG.
0574  32                    PUL A                   ;PULL AND
0575  A7  00                STA A   0,X             ;   SAVE THE FLOATING POINT NUMBER.
0577  32                    PUL A                   ;       X
0578  A7  01                STA A   1,X             ;       X
0579  32                    PUL A                   ;       X
057B  A7  02                STA A   2,X             ;       X
057D  32                    PUL A                   ;       X
057E  A7  03                STA A   3,X             ;       X
0580  32                    PUL A                   ;       X
0581  A7  04                STA A   4,X             ;       X
0583  32                    PUL A                   ;       X
0584  A7  05                STA A   5,X             ;       X
0586  32                    PUL A                   ;       X
0587  A7  06                STA A   6,X             ;     X X X
0589  32                    PUL A                   ;     X X X
058B  A7  07                STA A   7,X             ;      XXX
058C  7E  0000G             JMP     DREXTB          ;       X
```

```
                                        .SBTTL  TMULT   MULTIPLIES X5-X0 BY 10
                                        ;
                                        ;ROUTiEN TO MULT. X5-X0 BY 10, OVERFLOW TO B REG
                                        ;
058F  5F                        TMULT:  CLR B
0590  36                                PSH A           ;SAVE A TO ADD TO RESULT LATTER
0591  96  00G                           LDA A   X0,D
0593  48                                ASL A
0594  97  00G                           STA A   Y0,D
0596  96  00G                           LDA A   X1,D
0598  49                                ROL A
0599  97  00G                           STA A   Y1,D
059B  96  00G                           LDA A   X2,D
059D  49                                ROL A
059E  97  00G                           STA A   Y2,D
05A0  96  00G                           LDA A   X3,D
05A2  49                                ROL A
05A3  97  00G                           STA A   Y3,D
05A5  96  00G                           LDA A   X4,D
05A7  49                                ROL A
05A8  97  00G                           STA A   Y4,D
05AA  96  00G                           LDA A   X5,D
05AC  49                                ROL A
05AD  97  00G                           STA A   Y5,D
05AF  59                                ROL B
05B0  78  0000G                         ASL     Y0
05B3  79  0000G                         ROL     Y1
05B6  79  0000G                         ROL     Y2
05B9  79  0000G                         ROL     Y3
05BC  79  0000G                         ROL     Y4
05BF  79  0000G                         ROL     Y5
05C2  59                                ROL B           ;Y=X*4
05C3  96  00G                           LDA A   Y0,D
05C5  98  00G                           ADD A   X0,D
05C7  97  00G                           STA A   X0,D
05C9  96  00G                           LDA A   Y1,D
05CB  99  00G                           ADC A   X1,D
05CD  97  00G                           STA A   X1,D
05CF  96  00G                           LDA A   Y2,D
05D1  99  00G                           ADC A   X2,D
05D3  97  00G                           STA A   X2,D
05D5  96  00G                           LDA A   Y3,D
05D7  99  00G                           ADC A   X3,D
05D9  97  00G                           STA A   X3,D
05DB  96  00G                           LDA A   Y4,D
05DD  99  00G                           ADC A   X4,D
05DF  97  00G                           STA A   X4,D
05E1  96  00G                           LDA A   Y5,D
05E3  99  00G                           ADC A   X5,D
05E5  97  00G                           STA A   X5,D
05E7  C9  00                            ADC B   0,I     ;X=X*5
05E9  78  0000G                         ASL     X0
05EC  79  0000G                         ROL     X1
05EF  79  0000G                         ROL     X2
05F2  79  0000G                         ROL     X3
05F5  79  0000G                         ROL     X4
05F8  79  0000G                         ROL     X5
```

```
   05FB  59                 ROL B          ;X=X*10
   05FC  32                 PUL A
   05FD  98   00G           ADD A  X0,D            ;ADD ACCA TO X
   05FF  97   00G           STA A  X0,D
   0601  24   1A            BCC    1$
   0603  7C   0000G         INC    X1
   0606  26   15            BNE    1$
   0608  7C   0000G         INC    X2
   060B  26   10            BNE    1$
   060D  7C   0000G         INC    X3
   0610  26   08            BNE    1$
   0612  7C   0000G         INC    X4
   0615  26   06            BNE    1$
   0617  7C   0000G         INC    X5
   061B  26   01            BNE    1$
   061C  5C                 INC B
   061D  39            1$:  RTS
```

```
                                        .SBTTL  CMPFPN   COMPARES TWO FLOATING POINT NUMBERS
                                        ;
                                        ;
                                        ;THIS SUBROUTINE COMPARES THE POSITIVE FLOATING POINT NUMBER (FPN)
                                        ;       ON STACK(X) TO THE FPN THAT  TABPNT POINTS TO.
                                        ;RETURN WITH A=0 IF SAME
                                        ;           A=1 IF X GREATER THAN
                                        ;           A=-1 IF S LESS THAN
                                        ;
061E  DE  000G    CMPFPN: LDX     TABPNT,D
0620  A6  00              LDA A   0,X             ;GET EXP BYTES OF POINTED NO.
0622  E6  01              LDA B   1,X
0624  30                  TSX
0625  A0  07              SUB A   3,X
0627  25  3F              BCS     XGT
0629  26  3A              BNE     XLT             ;AND COMPARE TO STACK
062B  E1  04              CMP B   4,X
062D  25  39              BCS     XGT             ;"X>" TABLE.
062F  26  34              BNE     XLT             ;"X<" TABLE.
0631  A6  05              LDA A   5,X             ;  2 BYTES OF MANTISSA
0633  E6  06              LDA B   6,X             ;       OF X.
0635  DE  000G            LDX     TABPNT,D
0637  E0  03              SUB B   3,X
0639  A2  02              SBC A   2,X
063B  25  28              BCS     XLT             ;"X<" TABLE.
063D  26  29              BNE     XGT             ;"X>" TABLE
063F  5D                  TST B
0640  26  26              BNE     XGT             ;"X>" TABLE.
0642  E6  05              LDA B   5,X             ;GET THE NEXT TWO
0644  A6  04              LDA A   4,X             ;   BYTES OF THE MANTISSA.
0646  30                  TSX
0647  E0  08              SUB B   8,X
0649  A2  07              SBC A   7,X
064B  25  18              BCS     XGT             ;"X>" TABLE.
064D  26  16              BNE     XLT             ;"X<" TABLE.
064F  5D                  TST B
0650  26  13              BNE     XLT             ;"X<" TABLE.
0652  E6  0A              LDA B   10.,X           ;GET THE LAST TWO
0654  A6  09              LDA A   9.,X            ;   BYTES OF THE MANTISSA.
0656  DE  000G            LDX     TABPNT,D
0658  E0  07              SUB B   7,X
065A  A2  06              SBC A   6,X
065C  25  07              BCS     XLT             ;"X<" TABLE
065E  26  08              BNE     XGT             ;"X>" TABLE.
0660  5D                  TST B
0661  26  05              BNE     XGT             ;"X>" TABKE.
0663  4F                  CLR A
0664  39                  RTS                     ;BYE.....
0665  86  FF      XLT:    LDA A   377,1
0667  39                  RTS                     ;BYE.....
0668  86  01      XGT:    LDA A   1,1
066A  39                  RTS                     ;BYE.....
```

```
                                        .SBTTL  FUZ     FUZZY COMPARE

0668    BD      0000G   FUZZIE: JSR     PSHRET

                        ;CHECK FOR EITHER ARGUMENT=0
066E    A6      03              LDA A   3,X
0670    27      04              BEQ     1$
0672    A6      0C              LDA A   12.,X
0674    26      1E              BNE     NONZ            ;NEITHER IS
0676    BD      009C'   1$:     JSR     FPSUB           ;ONE OF THEM IS 0,SUB
0679    CE      0000G           LDX     #FUZB.1
067C    DF      00G             STX     TABPNT,D                ;AND CHECK AGAINST FUZB
067E    30                      TSX
067F    A6      01              LDA A   1,X
0681    97      00G             STA A   T4,D
0683    84      07              AND A   #7,1            ;ABS VALUE
0685    A7      01              STA A   1,X
0687    BD      061E'           JSR     CMPFPN
068A    2F      16              BLE     SAMEF           ;A=1 IF NO COMPARE
068C    96      00G             LDA A   T4,D
068E    2A      1A              BPL     AGT             ;SEE WHO WAS GREATER
0690    86      FF      BGT:    LDA A   #-1,1
0692    20      18              BRA     CLEAN
0694    7F      0000G   NONZ:   CLR     FUZR            ;SUB, AND CHECK INFO BACK FROM
                                                        ;THE NORMALIZE ROUTIEN
0697    BD      009C'           JSR     FPSUB
069A    B6      0000G           LDA A   FUZR
069D    B1      0000G           CMP A   FUZA
06A0    25      03              BCS     NOTSAM          ;ENOUGH BITS COMPARE
06A2    4F              SAMEF:  CLR A
06A3    20      07              BRA     CLEAN
06A5    30              NOTSAM: TSX
06A6    A6      01              LDA A   1,X     ;CHECK WHO WAS GREATER
06A8    2B      E5              BMI     BGT
06AA    86      01      AGT:    LDA A   #1,1
06AC    97      00G     CLEAN:  STA A   T4,D
06AE    7E      021A'           JMP     ZANS    ;AND LEAVE A 0 ON STACK
```

```
                                        .SBTTL  FPNASC   FLOATING POINT TO ASCII CONVERSION
                                        ;
                                        ;ROUTINE TO CONVERT FLOATING POINT NUMBER TO AN ASCII STRING.
                                        ;       REDUCES RANGE AND LEAVES DECREMENTED EXPONENT
                                        ;       AND BINARY FRACTION.
                                        ;
0681  BD  0000G   FPNASC: JSR     PSHRET
0684  A6  03              LDA A   3,X             ;UPPER MANTISSA
0686  B7  0000G           STA A   N12
0689  26  07              BNE     1$
068B  BD  0000G           JSR     A9X             ;THE NUMBER IS ZERO
068E  35                  TXS                     ;PRUNE
068F  7E  0000G           JMP     RTRN            ;RETURN.
06C2  C6  20      1$:     LDA B   40.I            ;X<>0
06C4  A6  01              LDA A   1,X             ;SIGN + UPPER EXPONENT.
06C6  2A  06              BPL     2$
06C8  C6  2D              LDA B   '-.I            ;NEGITIVE
06CA  88  80              EOR A   200,I           ;ABSOLUTE VALUE.
06CC  A7  01              STA A   1,X
06CE  E7  0000G   2$:     STA B   NS              ;SIGN OF NUMBER
06D1  5F                  CLR B
06D2  D7  00G             STA B   DEXP,D
06D4  D7  01G             STA B   DEXP+1,D
06D6  C6  06              LDA B   6,I
06D8  D7  00G             STA B   T4,D
06DA  81  04              CMP A   4,I             ;CHECK EXPONENT
06DC  2B  6A              BMI     LTONE           ;X/<1
06DE  26  0A              BNE     GTONE           ;X > 1
06E0  6D  02              TST     2,X
06E2  26  06              BNE     GTONE
06E4  86  FF              LDA A   377,I
06E6  16                  TAB                     ;DEXP=-1
06E7  7E  0791'           JMP     SETPUL          ;0.5 <= X < 1, NO REDUCTION
06EA  CE  0830'   GTONE:  LDX     PLTAB,I
06ED  DF  00G             STX     TABPNT,D        ;COMPARE TO TABLE OF
                                        ;   POWERS OF  10.
06EF  CE  0898'           LDX     NGTAB,I
06F2  DF  00G     1$:     STX     MULPNT,D        ;MULTIPLY BY NEG. POWER OF 10.
06F4  BD  061E'           JSR     CMPFPN          ;COMPARE TWO NUMBERS
06F7  4A                  DEC  A
06F8  26  08              BNE     2$              ;IF X.) 10.N
06FA  DE  00G             LDX     MULPNT,D        ;MULTIPLY BY 10 N
06FC  BD  0549'           JSR     PSHFPN
06FF  BD  030C'           JSR     FPMUL
0702  7C  0001G           INC     DEXP+1          ;NOTE IN EXPONENT
0705  78  0001G   2$:     ASL     DEXP+1
0708  8D  36              BSR     UPD1            ;COMMON ROUTINE
070A  7A  0000G           DEC     T4
070D  27  07              BEQ     3$              ;6 CONSTANTS TRIED
070F  DE  00G             LDX     MULPNT,D        ;UPDATE MULPNT NOW
0711  BD  0000G           JSR     A8X
0714  20  0C              BRA     1$
0716  86  07      3$:     LDA A   7,I             ;SET COUNTER
0718  97  00G             STA A   T4,D
071A  BD  061E'   10$:    JSR     CMPFPN          ;  10 8 >= X >= 1
071D  4A                  DEC A
071E  27  07              BEQ     4$              ;IF X>10 N, MULT BY 10 -(N+1)
```

```
0720  8D  17                    BSR    UPDATE
0722  7A  0000G                 DEC    T4
0725  26  F3                    BNE    10$
0727  DE  00G         4$:       LDX    MULPNT,D       ;FOUND RIGHT ONE!!!
0729  BD  0549'                 JSR    PSHFPN
072C  BD  030C'                 JSR    FPMUL          ;MULTIPLY FPN
072F  96  01G                   LDA A  DEXP+1,D                 ;FIX EXPONENT
0731  5F                        CLR B
0732  48                        ASL A
0733  48                        ASL A
0734  59                        ROL B
0735  9B  00G                   ADD A  T4,D           ; 0.1<= X <= 1.0
0737  20  58                    BRA    SETPUL
                      ;COMMON ROUTINE USED HEREABOUTS
0739  DE  00G         UPDATE:   LDX    MULPNT,D
073B  BD  0000G                 JSR    A8X            ;INC INDEX REG BY 8
073E  DF  00G                   STX    MULPNT,D
0740  DE  00G         UPD1:     LDX    TABPNT,D
0742  BD  0000G                 JSR    A8X            ;INC INDEX REG BY 8
0745  DF  00G                   STX    TABPNT,D
0747  39                        RTS
0748  CE  0830'       LTONE:    LDX    PLTAB,I
074B  DF  00G                   STX    MULPNT,D       ;MULTIPLY BY NUMBERS > 1
074D  CE  0898'                 LDX    NGTAB,I
0750  DF  00G                   STX    TABPNT,D       ;CMP TO NUMBERS < 1
0752  BD  061E'       1$:       JSR    CMPFPN
0755  4A                        DEC A
0756  2A  08                    BPL    2$
0758  DE  00G                   LDX    MULPNT,D       ; X <= 10 (-N)
075A  BD  0549'                 JSR    PSHFPN         ;      SO
075D  BD  030C'                 JSR    FPMUL          ;      MULTIPLY BY 10 N
0760  7C  0001G                 INC    DEXP+1
0763  8D  D4          2$:       BSR    UPDATE
0765  78  0001G                 ASL    DEXP+1
0768  7A  0000G                 DEC    T4             ;COUNT DOWN
076B  26  E5                    BNE    1$             ; GO AGAIN
076D  86  07                    LDA A  7,I            ;1 > X >= 10 8
076F  97  00G                   STA A  T4,D           ;7 CONSTANTS TO TRY
0771  BD  061E'       3$:       JSR    CMPFPN
0774  2F  09                    BLE    4$             ; 10 -N  >= X, SO MULT BY 10 N
0776  8D  C1                    BSR    UPDATE
0778  7A  0000G                 DEC    T4             ;DONE WITH
077B  26  F4                    BNE    3$             ;    SEARCH? IF NO, LOOP.
077D  20  08                    BRA    5$             ;MULT. BY 10 0 (NOPERATION)

077F  DE  00G         4$:       LDX    MULPNT,D
0781  BD  0549'                 JSR    PSHFPN
0784  BD  030C'                 JSR    FPMUL          ;DO FLOATING POINT MULTIPLY
0787  96  01G         5$:       LDA A  DEXP+1,D
0789  5F                        CLR B
078A  48                        ASL A
078B  48                        ASL A
078C  59                        ROL B                 ;MAKE EXPONENT
078D  9B  00G                   ADD A  T4,D
078F  43                        COM A
0790  53                        COM B                 ;   THE NEG OF EXP - 1
0791  97  01G         SETPUL:   STA A  DEXP+1,D
```

```
 115  0793  D7   00G            STA B  DEXP,D
 116  0795  CE   FEFEG  PULLER: LDX    X5-2,I
 117  0798  BD   0560'          JSR    PULFPN          ;MOVE TO Y1, Y0, X5>X0
 118  079B  96   00G            LDA A  Y0,D            ;SHIFT SO BINARY POINT IS
 119  079D  40                  NEG A                  ;  AT LEFT OF X5
 120  079E  CE   FFFDG          LDX    X5-3,I
 121  07A1  BD   0078'          JSR    SHMR            ;SHIFT
 122  07A4  7E   0000G  RETF:   JMP    RTRN
```

```
                                        .SBTTL  RNDATA   ROUNDS FPN AND PRODUCES ASCII STRING
                                        ;
                                        ;ROUTINE TO ROUND A FLOATING POINT NUMBER AND PRODUCES A
                                        ;       AN ASCII STRING.   CONVERTS EXPONENT TO AN ASCII STRING.
                                        ;
07A7  97    00G     RNDATA: STA A   T4,D            ;NUMBER OF DIGITS
07A9  7F    0000G           CLR     T1
07AC  CE    0000G           LDX     N12,I
                                        ;
07AF  86    28              LDA A   40.,I           ;ADD IN FUZZ SO ROUND UP WILL OCOUR
07B1  BD    058F'           JSR     TMULT           ;GET DIGIT
07B4  C1    0A              CMP B   10.,I           ;ROUND UP?
07B6  26    06              BNE     1$
07B8  20    2F              BRA     9$
07BA  4F            2$:     CLR A
07BB  BD    058F'                   JSR     TMULT           ;NEXT DIGIT
07BE  CB    30      1$:     ADD B   '0,I            ;MAKE ASCII
07C0  E7    00              STA B   0,X             ;STORE IT
07C2  09                    DEX                     ;POINT TO NEXT
07C3  7C    0000G           INC     T1
07C6  7A    0000G           DEC     T4
07C9  26    EF              BNE     2$              ;MORE TO GO
07CB  96    00G             LDA A   X5,D            ;TEST FOR ROUNDING
07CD  2B    0C              BMI     10$
07CF  86    30              LDA A   '0,I            ;NO ROUND, COUNT TRAILING 0'S
07D1  A1    01      3$:     CMP A   1,X
07D3  26    21              BNE     CEX             ;NOT 0, GO ON
07D5  7A    0000G           DEC     T1              ;0, ONE LESS DIGIT
07D8  08                    INX
07D9  20    F6              BRA     3$
                                        ;
07DB  86    3A      10$:    LDA A   72,I            ;  '9 + 1
07DD  6C    01      11$:    INC     1,X
07DF  A1    01              CMP A   1,X             ;KEEP ROUNDING
07E1  26    13              BNE     CEX
07E3  08                    INX
07E4  7A    0000G           DEC     T1
07E7  26    F4              BNE     11$
07E9  DE    00G     9$:     LDX     DEXP,D          ;ROUNDED ALL THE WAY
07EB  08                    INX
07EC  DF    00G             STX     DEXP,D          ;INCREMENT EXPONENT
07EE  7C    0000G           INC     T1              ;SET ANS TO 1 DIGIT OF 1
07F1  C6    31              LDA B   61,I
07F3  F7    0000G           STA B   N12
                                        ;
07F6  86    30      CEX:    LDA A   '0,I            ;ZERO EXPONENT
07F8  B7    0000G           STA A   E3              ;  UPPER DIGIT
07FB  C6    2B              LDA B   '+,I            ;MAKE SIGN
07FD  F7    0000G           STA B   ES              ;ASSUME POSATIVE
0800  D6    01G             LDA B   DEXP+1,D        ;GET DEXP INTO AB
0802  96    00G             LDA A   DEXP,D
0804  2A    08              BPL     2$              ;ALREADY PLUS
0806  C6    2D              LDA B   '-,I
0808  F7    0000G           STA B   ES              ;SET SIGN
080B  4F                    CLR A
080C  5F                    CLR B
080D  D0    01G             SUB B   DEXP+1,D        ;ABS. VALUE
```

```
    58  080F  92  00G          SBC A  DEXP,D
    59  0811  C0  64    2$:    SUB B  100.,I      ;DIVIDE THE DUMP AWAY
    60  0813  82  00           SBC A  0,I
    61  0815  25  05           BCS    3$          ;DON'T GO
    62  0817  7C  0000G        INC    E3
    63  081A  20  F5           BRA    2$
    64
    65  081C  CB  64    3$:    ADD B  100.,I      ; 0 <= B <= 99
    66  081E  86  2F           LDA A  '0-1,I
    67  0820  4C        4$:    INC A
    68  0821  C0  0A           SUB B  10.,I
    69  0823  24  F8           BCC    4$
    70  0825  87  0000G        STA A  E2
    71  0828  CB  3A           ADD B  '0+10.,I    ;OFF BY 10
    72  082A  F7  0000G        STA B  E1
    73  082D  96  00G          LDA A  T1,D        ;SET P.C-A
    74  082F  39               RTS                ;BYE
```

```
                                        .SBTTL  POWERS OF TEN - CONSTANTS

                                        ;
                                        ;********* FLOATING POINT CONSTANT, CONVERSION TABLE **********

                                        ;          NUMBER FORMAT IS   EE.HHHHHH
                                        ;                 WHERE EE     IS EXPONENT AS A POWER OF TWO
                                        ;                              EXPRESSED AS TWO HEXITS
                                        ;                      .       IS BINARY POINT.
                                        ;                 HHHHHH       IS BINARY FRACTION EXPRESSED AS 6 HEXITS
                                        ;
             0830   07    53    AA  PLTAB:  .BYTE   H07, H53, H0AA, H7E, H0EB, H0FB, H90, H0F9 ; 10 256
      0833   7E     EB    FB
      0836   90     F9
             0838   05    AA    93          .BYTE   H05, H0AA, H93, H0BA, H47, H0C9, H80, H0E9 ; 10 128
      083B   BA     47    C9
      083E   80     E9
             0840   04    D5    C2          .BYTE   4, H0D5, H0C2, H78, H1F, H49, H0FF, H0CF ; 10 64
      0843   78     1F    49
      0846   FF     CF
             0848   04    6B    9D          .BYTE   4, H6B, H9D, H0C5, H0AD, H0A8, H2B, H70 ; 10 32
      084B   C5     AD    A8
      084E   2B     70
             0850   04    36    8E          .BYTE   4, H36, H8E, H1B, H0C9, H0BF, H04,0     ; 10 16
      0853   1B     C9    BF
      0856   04     00
             0858   04    1B    BE          .BYTE   4, H1B, H0BE, H0BC, H20, 0,  0, 0       ; 10 8
      085B   BC     20    00
      085E   00     00
             0860   04    18    98          .BYTE   4. H18, H98, H96, H9C,  0,  0,  0      ; 10 7
      0863   96     80    00
      0866   00     00
             0868   04    14    F4          .BYTE   4, H14, H0F4, H24,  0,   0,  0, 0      ; 10 6
      086B   24     00    00
      086E   00     00
             0870   04    11    C3          .BYTE   4, H11, H0C3, H50,  0,   0,  0, 0      ;10 5
      0873   50     00    00
      0876   00     00
             0878   04    0E    9C          .BYTE   4, H0E, H9C, H40,   0,   0,  0, 0      ; 10 4
      087B   40     00    00
      087E   00     00
             0880   04    0A    FA          .BYTE   4, H0A, H0FA, 0,    0,   0,  0, 0      ; 10 3
      0883   00     00    00
      0886   00     00
             0888   04    07    C8          .BYTE   4, H07, H0C8,  0,   0,   0,  0, 0      ; 10 2
      088B   00     00    00
      088E   00     00
             0890   04    04    A0          .BYTE   4, H04, H0A0,  0,   0,   0,  0, 0      ; 10 1
      0893   00     00    00
      0896   00     00


             0898   00    AE    C0  NGTAB:  .BYTE   H00, H0AE, H0C0, H31, H43, H25, H63, H7A  ; 10 -256
      089B   31     43    25
      089E   63     7A
JJJJJ
             08A0   02    57    DD          .BYTE   H02, H57, H0DD, H0DD, H46, H7C, H64, H0BC ; 10 -128
```

```
                08A3    00      46      7C
                08A6    64      BC
     30         08A8    03      2C      A8          .BYTE   H03, H2C, H0A8, H7F, H0EA, H27, H0A5, H39 ; 10 -64
                08AB    7F      EA      27
                08AE    A5      39
     31         08B0    03      96      CF          .BYTE   H03, H96, H0CF, H081, H1E, H0AD, H45, H39 : 10 -32
                08B3    81      1E      AD
                08B6    45      39
     32         08B8    03      CB      E6          .BYTE   H03, H0CB, H0E6, H95, H94, H08E, H0C4, H4D ; 10 -16
                08BB    95      94      BE
                08BE    C4      4D
     33         08C0    03      E6      A8          .BYTE   H03, H0E6, H0A8, H0CC, H77, H11, H84, H61  ; 10 -8
                08C3    CC      77      11
                08C6    84      61
     34         08C8    03      E9      D6          .BYTE   H03, H0E9, H0D6, H08F, H94, H0D5, H0E5, H7A ; 10 -7
                08CB    8F      94      D5
                08CE    E5      7A
     35         08D0    03      E0      86          .BYTE   H03, H0ED, H86, H37, H08D, H05, H0AF, H6C   ; 10 -6
                08D3    37      8D      05
                08D6    AF      6C
     36         08D8    03      F0      A7          .BYTE   H03, H0F0, H0A7, H0C5, H0AC, H47, H18, H47  ; 10 -5
                08DB    C5      AC      47
                08DE    18      47
     37         08E0    03      F3      D1          .BYTE   H03, H0F3, H0D1, H087, H17, H58, H0E2, H19  ; 10 -4
                08E3    87      17      58
                08E6    E2      19
     38         08E8    03      F7      83          .BYTE   H03, H0F7, H83, H12, H6E, H97, H8D, H4F     ; 10 -3
                08EB    12      6E      97
                08EE    8D      4F
     39         08F0    03      FA      A3          .BYTE   H03, H0FA, H0A3, H0D7, H0A, H3D, H70, H0A3  : 10 -2
                08F3    D7      0A      3D
                08F6    70      A3
     40         08F8    03      FD      CC          .BYTE   H03, H0FD, H0CC, H0CC, H0CC, H0CC, H0CC, H0CC ; 10 -1
                08FB    CC      CC      CC
                08FE    CC      CC
     41
     42
     43                 0001'                       .END
```

SYMBOL TABLE

```
ADD     0372R   02 AGAIN   034CR   02 AGT     06AAR   02 ALIGN   000DR   02 ASCDN   052CR   02
ASCD1   0532R   02 ASCFPN  07CDRG  02 A10X  = XXXXXX G     A11X  = XXXXXX G     A12X  = XXXXXX G
A13X  = XXXXXX G     A5X   = XXXXXX G     A6X   = XXXXXX G     A7X   = XXXXXX G     A8X   = XXXXXX G
A9X   = XXXXXX G     BANS    0086R   02 BANS1   0006R   02 BGT     0690R   02 CEX     07F6R   02
CHAR  = XXXXXX G     CLEAN   069CR   02 CMPFPN  0?1ERG  02 CNT   = XXXXXX G     DEXP  = XXXXXX G
DIGFLG= XXXXXX G     DREXTB= XXXXXX G     EFLAG =   'X G     EQU   = XXXXXX G     ERFIXN= XXXXXX G
ERFPDV= XXXXXX G     ERFPOV= XXXXXX G     ERFXOV= X   .X G     ERRCD = XXXXXX G     ES    = XXXXXX G
F1    = XXXXXX G     F2    = XXXXXX G     F3    = XX  XX G     FIXRN1  0044R   02 FIXRTN  0042R   02
FIX1    0000RG  02 FIX1A   0004R   02 FLOAT1  0059RG  02 FLOAT2  0059RG  02 FPADD   00A3RG  02
FPDIV   0231RG  02 FPMUL   030CRG  02 FPNASC  0681RG  02 FPRET   021DR   02 FPSUB   009CRG  02
FRTST   01E1R   02 FUZA  = XXXXXX G     FUZB  = XXXXXX G     FUZR  = XXXXXX G     FUZZIE  0668RG  02
GETCHR= XXXXXX G     GTONE   06EAR   02 ITSINT= XXXXXX G     LPABC   0519R   02 LTONE   0748R   02
MANTLP  0407R   02 MANTL1  040AR   02 MAXANS  01FFRG  02 MOV     028AR   02 MUP     04F5R   02
MUP1    04E8R   02 MULEXT= XXXXXX G     MURETU  0396R   02 NEGERR  0045R   02 NEXT    0370R   02
NEXTBY  03ABR   02 NGTAB   0898RG  02 NONZ    0694R   02 NORM    01A1RG  02 NORM.   0187RG  02
NORM1   01A3R   02 NORM2   01E1R   02 NORM3   018CR   02 NORM4   01D4R   02 NOTINT  0537R   02
NOTSAM  06A5R   02 NS    = XXXXXX G     N12   = XXXXXX G     OERR    0040R   02 OVER    01FBR   02
PLTAB   0830RG  02 PSHFPN  0549RG  02 PSHRET= XXXXXX G     PULFPN  056DRG  02 PULLER  0795R   02
RETF    07A4R   02 RETS    04D8R   02 RETURN  0382R   02 RNDATA  07A?RG  02 RTRN  = XXXXXX G
SAMEF   06A2R   02 SETPUL  0791R   02 SHLP    006ER   02 SHMR    007BRG  02 SIGNS = XXXXXX G
TABPNT= XXXXXX G     TFLGS1= XXXXXX G     TMULT   058FR   02 TRYE    043DR   02 TWO     048DR   02
T1    = XXXXXX G     T2    = XXXXXX G     T3    = XXXXXX G     T4    = XXXXXX G     UNDER   021ARG  02
UPDATE  0739R   02 UPD1    0740R   02 WALTG = XXXXXX G     XGT     0668R   02 XLT     0665R   02
X0    = XXXXXX G     X1    = XXXXXX G     X2    = XXXXXX G     X3    = XXXXXX G     X4    = XXXXXX G
X5    = XXXXXX G     Y0    = XXXXXX G     Y1    = XXXXXX G     Y2    = XXXXXX G     Y3    = XXXXXX G
Y4    = XXXXXX G     Y5    = XXXXXX G     ZANS    021ARG  02 ZFIX    007ER   02 ZIX     0220RG  02
ZIX4    0228RG  02
. ABS.  0000    00
        0000    01
FPMATH  0900    02
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2687. WORDS
.SY:FPMATH/C(DK1:SETCL),FPMATH
```

CROSS REFERENCE TABLE (CREF V01-03)

```
A10X    1-31#
A11X    1-27#
A12X    1-31#
A13X    1-31#
A5X     1-27#
A6X     1-27#
A7X     1-27#
A8X     1-27#   7-181   7-197   13-51   13-73   13-76
A9X     1-25#   3-32    3-60    3-109   3-153   5-26    5-33    6-20    13-11
ADD     6-73#
AGAIN   6-48#   6-102
AGT     12-21   12-35#
ALIGN   3-15    3-35#
ASCD1   7-188   7-197#
ASCDN   7-186   7-193   7-195#
ASCFPN  1-25#   7-10#
BANS    3-16#   3-49    3-51
BANS1   3-13    3-32#   3-62    3-64
BGT     12-22#  12-34
CEX     14-27   14-35   14-46#
CHAR    1-28#   7-37
CLEAN   12-23   12-31   12-36#
CMPFPN  1-32#   11-10#  12-18   13-39   13-55   13-83   13-96
CNT     1-20#   7-22    7-202
DEXP    1-28#   7-34#   7-99    7-102   7-108   7-110#  7-112#  7-114#  7-129   7-130   7-132   7-133   7-146#  7-147#
        7-168#  7-172#  7-185   13-22#  13-23#  13-45#  13-46#  13-64   13-89#  13-91#  13-106  13-114#  13-115#  14-39
        14-41#  14-50   14-51   14-57   14-58
DIGFLG  1-26#   7-20#   7-59#
DREXTB  1-26#   8-3#    8-5#    8-24    9-9#    9-11#   9-29
E1      1-26#   14-72#
E2      1-26#   14-70#
E3      1-26#   14-47#  14-62#
EFLAG   1-31#   7-35#   7-81#   7-93#
EQU     1-20#   7-204#
ERFIXN  1-33#   1-96
ERFPOV  1-36#   5-28
ERFPOV  1-35#   4-60
ERFXOV  1-34#   1-100
ERRCD   1-21#   1-93    1-97#   1-101#  4-61#   5-29#   7-195#
ES      1-26#   14-49#  14-54#
FIX1    1-30#   1-54#
FIX1A   1-56#   1-99
FIXRN1  1-95#   2-39
FIXRTN  1-72    1-87    1-93#   1-105
FLOAT1  1-30#   2-9#
FLOAT2  1-30#   2-10#
FPADD   1-30#   3-7#
FPDIV   1-30#   5-23#   7-177   7-192
FPMUL   1-30#   6-12#   7-179   7-194   13-44   13-63   13-88   13-105
FPNRSC  1-32#   13-7#
FPRET   4-55    4-75    4-79#   5-35
FPSUB   1-30#   3-2#    12-10   12-26
FRTST   3-111   4-53#
FUZA    1-22#   12-28
FUZB    1-22#   12-11
FUZR    1-22#   4-21#   4-46#   12-24#  12-27
```

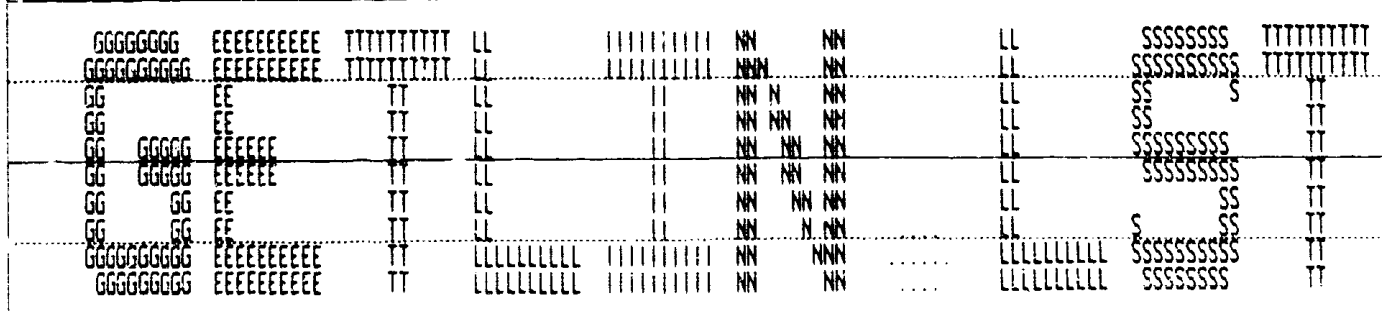
CROSS REFERENCE TABLE (CREF V01-03)

```
FUZZLE  1-23#  12-3#
GETCHR  1-28#  7-45    7-84    7-94
GTONE   13-28  13-30   13-34#
LISINT  1-19#  2-21*   2-200*
LPABC   7-187# 7-198
LTONE   13-27  13-79#
MANTL1  7-41   7-46#
MANTLP  7-35   7-45#   7-54    7-67    7-73
MAXANS  1-27#  4-62#   5-30    7-161
MOV     5-75#
MLP     7-172# 7-184
MLP1    7-169  7-173#
MULPNT  1-25#  13-38*  13-42   13-50   13-61   13-72   13-74*  13-80*  13-86   13-103
MURETU  6-72   6-91#
N12     1-26#  13-9*   14-8    14-44*
NEGERR  1-55   1-96#
NEXT    6-72#  6-99
NEXTBY  6-52   6-64    6-100#
NGTAB   1-23#  13-37   13-81   15-28#
NONZ    12-9   12-24#
NORM    1-25#  2-23    4-12#   4-25    5-109   6-120
NORM1   4-14#  4-37
NORM2   4-15   4-45#
NORM3   4-18   4-26#
NORM4   4-38#  4-45
NORMR   4-23#  4-24#   7-157
NOTINT  7-42   7-52    7-92    7-123   7-200#
NOTSAM  12-29  12-32#
NS      1-26#  13-20*
OERR    1-58   1-76    1-85    1-100#
OVER    4-60#
PLTAB   1-23#  7-166   13-34   13-79   15-13#
PSHFPN  1-32#  7-149   7-174   7-189   8-2#    13-43   13-62   13-87   13-104
PSHRET  1-29#  2-10    3-7     4-24    5-23    6-17    7-10    12-3    13-7
PULFPN  1-32#  9-8#    13-117
PULLER  13-116#
RETF    13-122#
RETS    7-158#
RETURN  6-108#
RNDATA  1-32#  14-6#
RTRN    1-29#  3-34    4-22    4-79    7-196   13-13   13-122
SAMEF   12-19  12-30#
SETPUL  13-33  13-70    13-114#
SHLP    2-29#  2-37
SHMR    1-28#  2-36#   3-67    13-121
SIGNS   1-24#  1-25#   7-36*   7-44*   7-91*   7-125   7-157
T1      1-24#  5-77*   5-99    5-117*  5-141   6-16*   6-47*   6-101*  7-32*   7-50    7-53*   7-60    14-7*   14-20*
        14-28* 14-37*  14-42*  14-73
T2      1-24#  5-76*   5-107*  6-51    6-66*   7-62    7-64*   7-70    7-72*   7-124
T3      1-24#  7-33*   7-127   7-134*  7-144   7-175   7-190
T4      1-28#  7-101*  7-105*  7-137*  7-145*  7-158   7-164*  7-183*  12-15*  12-20   12-36*  13-25*  13-48*  13-54*
        13-59* 13-69   13-92*  13-95*  13-99*  13-111  14-6*   14-21*
TABPNT  1-28#  7-167*  7-180   7-182*  11-10   11-22   11-40   12-12*  13-35*  13-75   13-77*  13-82*
TFLGS1  1-20#  7-25    7-27*   7-205   7-207*
TMULT   7-68   10-5#   14-11   14-16
TRUE    7-58   7-77#
TWO     7-49   7-51    7-80    7-96    7-98    7-118#
```

CROSS REFERENCE TABLE (CREF V01-03)

.INDEX	1-25#	4-57	4-76#											
.UPD1	13-47	13-75#												
.UPDATE	13-58	13-72#	13-90	13-98										
.WAITG	1-27#	2-21	8-22											
X0	1-21#	3-9#	3-28	3-77	5-72#	5-79	5-81#	5-110#	5-120	5-122#	6-45#	6-74	10-7	10-34
	10-35#	10-52#	10-60	10-61#										
X1	1-21#	5-70#	5-82	5-84#	5-111#	5-123	5-125#	6-43#	6-77	7-31#	10-10	10-37	10-38#	10-53#
	10-63#													
X2	1-21#	5-68#	5-85	5-87#	5-112#	5-126	5-128#	6-41#	6-80	7-121	10-13	10-40	10-41#	10-54#
	10-65#													
X3	1-21#	5-66#	5-88	5-90#	5-113#	5-129	5-131#	6-39#	6-83	7-30#	7-120	10-16	10-43	10-44#
	10-55#	10-67#												
X4	1-21#	5-64#	5-91	5-93#	5-114#	5-132	5-134#	6-37#	6-86	7-119	10-19	10-46	10-47#	10-56#
	10-69#													
X5	1-21#	5-62#	5-94	5-96#	5-115#	5-135	5-137#	6-35#	6-89	7-29#	7-65	7-118	7-148	10-22
	10-49	10-50#	10-57#	10-71#	13-116	13-120	14-23							
XGT	11-15	11-18	11-26	11-28	11-34	11-44	11-46	11-51#						
XLT	11-16	11-19	11-25	11-35	11-37	11-43	11-49#							
Y0	1-24#	5-58#	5-80	5-121	6-54#	6-73	6-75#	6-96#	6-118	10-9#	10-26#	10-33	13-118	
Y1	1-24#	5-56#	5-83	5-124	6-15#	6-53	6-56#	6-76	6-78#	6-95#	6-116	10-12#	10-27#	10-36
Y2	1-24#	5-54#	5-86	5-127	6-55	6-58#	6-79	6-81#	6-94#	6-114	10-15#	10-28#	10-39	
Y3	1-24#	5-52#	5-89	5-130	6-14#	6-57	6-60#	6-82	6-84#	6-93#	6-112	10-18#	10-29#	10-42
Y4	1-24#	5-50#	5-92	5-133	6-59	6-62#	6-85	6-87#	6-92#	6-110	10-21#	10-30#	10-45	
Y5	1-24#	5-48#	5-95	5-136	6-13#	6-61	6-63#	6-88	6-90#	6-91#	6-108	10-24#	10-31#	10-48
ZANS	1-27#	4-77#	6-22	7-162	12-37									
ZFIX	1-57	1-91#												
ZIX	1-29#	4-78	4-80#											
ZIX4	4-2#	4-84#												

SEL 1-3%



```
                                        .IDENT  /MC0502/
                                        .GLOBL  GETLIN
                0000'                   GETLIN=.
                                        .TITLE  GETLIN
                                        .GLOBL  GETSMA,GETLAR
                                        .GLOBL  PGMPTR,LNNOTG,IMXTG,HCTRCS


                                ;       THERE ARE ACTUALLY TWO ROUTINES HERE "GETSMA" AND "GETLAR".
                                ;       GETSMA SEARCHES FOR THE NEXT SMALLER OR EQUAL LINE NUMBER.
                                ;       WHILE GETLAR SEARCHES FOR THE NEXT LARGER OR EQUAL LINE
                                ;       NUMBER.

                                ;       INPUT:
                                ;               GETSMA: LINE NUMBER ON THE STACK.
                                ;               GETLAR: LINE NUMBER ON THE STACK.
                                ;
                                ;       OUTPUT:
                                ;               GETSMA: ON THE TOP OF THE STACK A POINTER TO THE LINE
                                ;                       WHOSE LINE NUMBER IS THE GREATEST, LESS THAN OR
                                ;                       EQUAL TO THE INPUT LINE NUMBER.  NEXT DOWN
                                ;                       ON THE STACK, THE LINE NUMBER OF THAT LINE.
                                ;               GETLAR: ON THE TOP OF THE STACK A POINTER TO THE LINE
                                ;                       WHOSE LINE NUMBER IS THE SMALLEST, GREATER THAN
                                ;                       OR EQUAL TO THE INPUT LINE NUMBER.  NEXT
                                ;                       DOWN ON THE STACK, THE LINE NUMBER OF THAT LINE.

                                ; NOTE: THIS ROUTINE USES THE LAST 8 BYTES OF THE SCRATCH AREA

0000    7F  FFF0G       START:  CLR     <HCTRCS-3>
0003    7F  FFFEG               CLR     <HCTRCS-2>
0006    32                      PUL A
0007    33                      PUL B
0008    B7  FFF9G               STA A   HCTRCS-7        ; HCTRCS IS RET. ADDR
000B    F7  FFFAG               STA B   <HCTRCS-6>
000E    32                      PUL A
000F    32                      PUL A                   ; HCTRCS-5 IS LINE #
0010    B7  FFF8G               STA A   <HCTRCS-5>
0013    32                      PUL A
0014    B7  FFFCG               STA A   <HCTRCS-4>
0017    DE    00G               LDX     PGMPTR,D        ;
0019    26    05        TOP:    BNE     1$
0018    FE  FFF9G               LDX     HCTRCS-7        ; IS THERE A NEXT LINE?
001E    6E    02                JMP     2,X
0020    FF  FFF0G       1$:     STX     <HCTRCS-3>      ; IF SO IS IT THE ONE
0023    A6    07                LDA A   7,X             ; WE WANT
0025    B1  FFF8G               CMP A   <HCTRCS-5>
0028    25    12                BCS     NEXT
002A    22    08                BHI     QUIT
002C    A6    08                LDA A   8.,X
002E    B1  FFFCG               CMP A   <HCTRCS-4>
0031    25    09                BCS     NEXT
0033    22    02                BHI     QUIT
0035    20    33                BRA     STACK
0037    FE  FFF9G       QUIT:   LDX     HCTRCS-7        ; THEN
```

```
003A  6E  00                JMP    0,X          ; RETURN
003C  EE  03       NEXT:    LDX    3,X          ; IF NOT THERE YET GET
                                                ; THE NEXT LINE
003E  20  D9                BRA    TOP
0040  32           GETSMA:  PUL A
0041  33                    PUL B
0042  B7  FFFFG             STA A  HCTRCS-1
0045  F7  0000G             STA B  HCTRCS
0048  8D  B6                BSR    START
004A  20  16                BRA    PASSED
004C  20  1C                BRA    STACK        ;CHANGED ********
004E  32           GETLAR:  PUL A
004F  33                    PUL B
0050  B7  FFFFG             STA A  HCTRCS-1
0053  F7  0000G             STA B  HCTRCS
0056  8D  A8                BSR    START
0058  20  10                BRA    STACK
005A  7F  FFFDG             CLR    HCTRCS-3
005D  7F  FFFEG             CLR    HCTRCS-2
0060  20  08                BRA    STACK        ;CHANGED *******
0062  FE  FFFDG    PASSED:  LDX    HCTRCS-3
0065  EE  05                LDX    5,X
0067  FF  FFFDG             STX    HCTRCS-3     ;MOVED *******
006A  FE  FFFDG    STACK:   LDX    HCTRCS-3
006D  A6  08                LDA A  8.,X
006F  36                    PSH A
0070  A6  07                LDA A  7,X
0072  36                    PSH A
0073  86  00G               LDA A  LNNOTG.!
0075  36                    PSH A
0076  B6  FFFEG             LDA A  (HCTRCS-2)
0079  36                    PSH A
007A  B6  FFFDG             LDA A  (HCTRCS-3)
007D  36                    PSH A
007E  86  00G               LDA A  INXTG.!
0080  36                    PSH A
0081  FE  FFFFG             LDX    HCTRCS-1
0084  6E  00                JMP    0,X
          0001'             .END
```

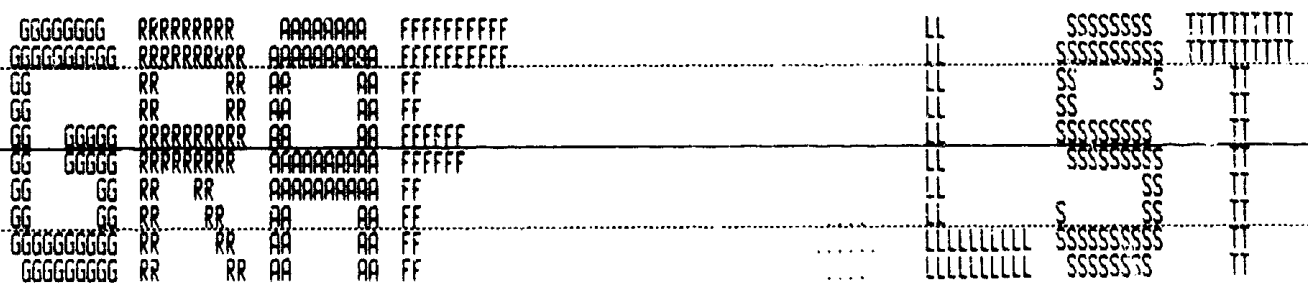
```
GETLIN RT-11 MMAC  VM02-10   14-OCT-76 01:33:33 PAGE 1+
SYMBOL TABLE

GETLAR    004ERG      GETLIN=   0000RG      GETSMA    0040RG      HCTRCS= ******  G      IMXTG = ******  G
LNNOTG= ******  G     NEXT      003CR       PASSED    0062R       PGMPTR= ******  G      QUIT      0037R
STACK     006AR       START     0000R       TOP       0019R
. ABS.    0000      00
          0086      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3425. WORDS
,SY:GETLIN/C(DK1:SE(CLI,GETLIN
```

CROSS REFERENCE TABLE (CREF V01-03)

1-18

GETLAR 1-20# 1-84#
GETLIN 1-17# 1-18#
GETSWI 1-20# 1-77#
HCTRCS 1-21# 1-47% 1-48% 1-51% 1-52% 1-55% 1-57% 1-60 1-62% 1-64 1-68 1-72 1-79% 1-80%
1-86% 1-87% 1-90% 1-91% 1-93 1-95% 1-96 1-103 1-105 1-109
INXTG 1-21# 1-107
LNNOTG 1-21# 1-101
NEXT 1-65 1-69 1-74#
PASSED 1-82 1-93#
PGMPTR 1-21# 1-58
QUIT 1-66 1-70 1-72#
STACK 1-71 1-83 1-89 1-92 1-96#
START 1-47# 1-81 1-88
TOP 1-59# 1-76

SEL 1-3#



GRAF ----THIS IS REALLY GRAPHIC RT-11 MMAC VM02-10 14-OCT-76 01:33:37
TABLE OF CONTENTS

```
                        .TITLE  GRAF ----THIS IS REALLY GRAPHICS
                        .GLOBL  GRAF
000000                  GRAF =  .
                        .SBTTL GRAPHICS SUBROUTINES
                        .IDENT  /SB0035/


                        ;MEMORY MAP,  TO BE ARRANGED AS LISTED

                        .GLOBL  GRFDR.
                        .GLOBL  QVIEW,QWINDOW,QSCALE,PLOTVL
                        .GLOBL  FPB,FPC         ;TEMP STORAGE
                .GLOBL XMINW,XMAXW,YMINW,YMAXW
                .GLOBL DLTXS,XMINS,DLTYS,YMINS,XSF,YSF
                .GLOBL XLAST,YLAST,SINTHA,COSTHA
                .GLOBL XNEW,YNEW,TEMPX,TEMPY
                        .GLOBL  GINSET          ; FOR ARROW TO USE

                                ;THE TABLE ENTRIES 0-3 ARE SET UP BY THE WINDOW COMMAND
                                ;(OR BY DEFAULT).   THE TABLE ENTRIES 4-7 ARE SET UP BY
                                ;THE VIEWPRT COMMAND (OR BY DEFAULT).
                                ;
                                ;USER UNITS ARE FPNS SET UP IN VIRTUAL SPACE, AND ARE
                                ;GIVEN AS ARGUMENTS IN MOVE, DRAW AND AXIS COMMANDS.
                                ;
                                ;GDUS ARE GRAPHIC DISPLAY UNITS IN SCREEN SPACE.
                                ;(130 GDUS = X WIDTH, 100 GDUS = Y WIDTH FOR DVST)

                                ;INDEX  NAME (DEFALT)   DISCRIPTION
                                ;0      XMINW   (0)     WINDOW (USER UNITS)
                                ;1      XMAXW   (130)   "
                                ;2      YMINW   (0)     "
                                ;3      YMAXW   (100)   "
                                ;4      DLTXS   (130)   VIEWPORT IN GDU (DELTA X )
                                ;5      XMINS   (0)     "               (X MIN )
                                ;6      DLTYS   (100)   "               (DELTA Y )
                                ;7      YMINS   (0)     "               (Y MIN )
                                ;8      XSF     (1)     AXIS SCALE FACTOR (USER UNITS/GDU)
                                ;9      YSF     (1)     "
                                ;10     XLAST   (0)     POSITION OF LAST POINT PLOTED
                                ;11     YLAST   (0)     "
                                ;12     SINTHA  (0)     RELATIVE ROTATE CONSTANT (SIN)
                                ;13     COSTHA  (1)     "                        (COS)
                                ;15     XNEW    (-)     NEW POSITION FOR PLOTING
                                ;16     YNEW    (-)     "
                                ;17     TEMPX   (-)     TEMP X VALUE
                                ;18     TEMPY   (-)     TEMP Y VALUE

                .GLOBL R0,R1,R2,R3,R4,R5,R6,R7   ;TEMP 2 BYTE REGISTERS
                        .GLOBL  ERRCD
                        .GLOBL  A.SEC
                        .GLOBL  DREXTA
                .GLOBL FPMUL  ;9 BYTE MATH ROUTINE
                .GLOBL PSHFPN,PULFPN  ;9 BYTE PUSH OR PULL THE STACK TO LOCATION GIVEN BY (X
                .GLOBL A8X    ;ADD 8 TO THE INDEX REGISTER
                        .GLOBL  A16X            ; GGUESS WHAT THAT DOES
                        .GLOBL  DOFP,FRET,FHEX,FHIN,FLEX,FLIN,FART
```

```
                                        .GLOBL  FSAV,FRES,FSWP
0016                                    FDUP    = ^H16
0080                                    FA      = 200
00A0                                    FS      = 240
00C0                                    FM      = 300
00E0                                    FD      = 340

                                        .GLOBL  TABPTR          ; USED AS X AXIS START OF ARRAY
                                        .GLOBL  LENGTH          ; USED AS X AXIS END OF ARRAY
                                        .GLOBL   COLCNT         ; USED A YAXIS START OF ARRAY
                                        .GLOBL  TCOL            ; USED AS Y AXIS END OF ARRAY
                                        .MACRO  SAVRTN          ; SAVE RETURN ADDRESS IN RETURN A
                                        PUL A
                                        STA A   DREXTP+1,D
                                        PUL A
                                        STA A   DREXTP+2,D
                                        .ENDM

                                        .MACRO  SET     K1,K2
                                        .BYTE   FHEX
                                        .WORD   K1
                                        .BYTE   FLEX
                                        .WORD   K2
                                        .ENDM
                                        .GLOBL  GRFINI          ; GRAFICS INITIALIZATION
                                MACRO   Q,K1
                                .BYTE   FLEX
                                .WORD   K1
                                .ENDM

                                .MACRO  R
                                .BYTE   FDUP
                                .ENDM

0000    BD      0000G           GRFINI: JSR     DOFP
0003    00G                             .BYTE   FHEX
0004    0000G                           .WORD   FPZERO
0006                                    R
0007                                    Q       XLAST
000A                                    R
000B                                    Q       YLAST
000E                                    R
000F                                    Q       SINTHA
0012                                    R
0013                                    Q       XMINW
0016                                    R
0017                                    Q       YMINW
001A                                    R
001B                                    Q       XMINS
001E                                    Q       YMINS
0021                                    SET     FPONE,COSTHA
0027    00G                             .BYTE   FHEX
0028    004F'                           .WORD   K.130
002A                                    R
002B                                    Q       XMAXW
002E                                    Q       DLTXS
0031    00G                             .BYTE   FHEX
```

```
          0032   0047'                     .WORD   K.100
          0034                              R
          0035                              Q       YMAXW
          0038                              Q       DLTYS
                                           .GLOBL  FPZERO,FPONE
          003B    00G                      .BYTE   FHEX
          003C   0000G                     .WORD   FPONE
          003E                              R
          003F                              Q       XSF
          0042                              Q       YSF
          0045    00G                      .BYTE   FRET
          0046    39                        RTS

          0047    04     07    C8  K.100:  .BYTE   4.,7.,200.,0.,0.,0.,0.,0
    004A    00    00     00
    004D    00    00
          004F    04     08    82  K.130:  .BYTE   4.,8.,130.,0,0,0,0,0
    0052    00    00     00
    0055    00    00
```

```
                                        .SBTTL  ROTATE--ROTATE DRIVER
                                ;THIS ROUTINE SETS UP COSTHA,SINTHA OF THE ROUTATE ANGLE
                                        .GLOBL  QROTATE
                                        .GLOBL  TYPARG,SETARG
                                        .GLOBL  SINCOD
                                        .GLOBL  COSCOD
                                        .GLOBL  CTKN
                                        .GLOBL  TRIG
                                        .GLOBL  ERUNOF
                                ;
0057                            QROTATE:        SAVRTN          ; SAVE RETURN ADDRESS
005D    BD      0000G                   JSR     SETARG          ; GET ARGUMENT ON STACK
0060    40                              TST A                   ; IS IT VALID?
0061    27      03                      BEQ     1$
0063    7E      0385'                   JMP     GRAFER          ; ALL OTHERS ARE ERRORS
0066    BD      0000G           1$:     JSR     DOFP            ; MAKE A COPY OF THE ANGLE
0069    16                              .BYTE   FDUP
006A    00G                             .BYTE   FRET
0068    86      00G                     LDA A   SINCOD,I        ; SET UP TO DO SIN
006D    97      00G                     STA A   CTKN,D
006F    BD      0000G                   JSR     TRIG
0072    CE      0000G                   LDX     SINTHA,I
0075    BD      0000G                   JSR     PULFPN          ; SET UP SINTHA
0078    86      00G                     LDA A   COSCOD,I        ; SET UP TO DO COS
007A    97      00G                     STA A   CTKN,D
007C    BD      0000G                   JSR     TRIG
007F    CE      0000G                   LDX     COSTHA,I        ; SET UP COSTHA
0082    BD      0000G                   JSR     PULFPN
0085    7E      0000G                   JMP     DREXTG
```

```
                                        .SBTTL ------------------------------------------------------------
                                        .SBTTL ROUTINES TO SET UP GRAPHICS PARAMETERS


                                        ;*****************************************************************************

                                        ;               ROUTINES TO SET UP GRAGHICS PRAMETERS

                                        ;*****************************************************************************

                                        .SBTTL SCALEF CALCULATE SCALE FACTOR PARAMETERS

                                        ; COMPUTE SCALE FACTORS
                                        ; CHANGE        XSF, YSF, RSF

0088    80   0000G     SCALEF: JSR     DOFP    ; XSF=(XMAXW - XMINW)/DLTXS
0088    00G                     .BYTE   FHEX
008C  0000G                     .WORD   XMAXW
008E    A0G                     .BYTE   FHEX+FS
008F  0000G                     .WORD   XMINW
0091    E0G                     .BYTE   FHEX+FD
0092  0000G                     .WORD   DLTXS
0094    16                      .BYTE   FDUP
0095    00G                     .BYTE   FHEX            ; UPDATE XLAST
0096  0000G                     .WORD   XLAST           ; NEW (= ((OLD- OLD MINW)/OLD SF)*NEW SF + NEW MINW
0098    A0G                     .BYTE   FHEX    +FS
0099  0000G                     .WORD   XNEW
0098    E0G                     .BYTE   FHEX+FD
009C  0000G                     XSF
009E    C0G                     .BYTE   FART + FM
009F    80G                     .BYTE   FHEX+FA
00A0  0000G                     XMINW
00A2    00G                     .BYTE   FLEX
00A3  0000G                     .WORD   XLAST
00A5    00G                     .BYTE   FLEX
00A6  0000G                     .WORD   XSF
                                ;YSF=(YMAXW-YMINW)/DLTYS
00A8    00G                     .BYTE   FHEX
00A9  0000G                     .WORD   YMAXW
00A8    A0G                     .BYTE   FHEX+FS
00AC  0000G                     .WORD   YMINW
00AE    E0G                     .BYTE   FHEX+FD
00AF  0000G                     .WORD   DLTYS
00B1    16                      .BYTE   FDUP            ; MAKE A COPY
00B2    00G                     .BYTE   FHEX
00B3  0000G                     YLAST
00B5    A0G                     .BYTE   FHEX+FS
00B6  0000G                     YNEW
00B8    E0G                     .BYTE   FHEX+FD
00B9  0000G                     YSF
00BB    C0G                     .BYTE   FART+FM
]]]]]
00BC    80G                     .BYTE   FHEX+FA
00BD  0000G                     YMINW
00BF    00G                     .BYTE   FLEX
00C0  0000G                     .WORD   YLAST
00C2    00G                     .BYTE   FLEX
00C3  0000G                     .WORD   YSF
```

SCALEF CALCULATE SCALE FACTOR PARAMETERS

```
58          00C5    00G                     .BYTE   FRET
59          00C6    2F  00000G              CLR     ERRCD
60          00C9    39                      RTS
61
```

```
                                .SBTTL WINDOW CALCULATE WINDOW PARAMETERS

                                ; ROUTINE TO SET WINDOW
                                ; STACK AT ENTRY YMAXW, YMINW, XMAXW  XMINW
                                ; WILL CHANGE XMINW, XMAXW, YMINW, YMAXW

00CA                    QWINDOW:        SAVRTN
0000  8D  4A                    BSR     GETIT
0002  26  3F                    BNE     QWINEX
0004  8D  0000G                 JSR     FPCMP
0007  40                        NEG A
0008  2B  3C                    BMI     WINERR          ; YMIN > YMAX CAUSES ERROR
000A  8D  0000G                 JSR     DOFP
000D                            SET     XMINW,XNEW
00E3                            SET     YMINW,YNEW
00E9  00G                       .BYTE   FLEX
00EA  0000G                     .WORD   TEMPX
00EC  00G                       .BYTE   FLEX
00ED  0000G                     .WORD   TEMPY
00EF  00G                       .BYTE   FRET
00F0  8D  2A                    BSR     GETIT
00F2  26  1F                    BNE     QWINEX
00F4  8D  0000G                 JSR     FPCMP
00F7  40                        NEG A
00F8  2B  1C                    BMI     WINERR          ; XMIN > XMAX CAUSES ERROR
00FA  8D  0000G                 JSR     DOFP
00FD  00G                       .BYTE   FLEX
00FE  0000G                     .WORD   XMAXW
0100  00G                       .BYTE   FLEX
0101  0000G                     .WORD   XMINW
0103                            SET     TEMPX,YMAXW
0109                            SET     TEMPY,YMINW
010F  00G                       .BYTE   FRET
0110  8D  0088'                 JSR     SCALEF
0113  7E  0000G         QWINEX: JMP     DREXTA
0116  86  00G           WINERR: LDA A   ERDOMN,I        ; SET ERROR
0118  97  00G                   STA A   ERRCD,D
011A  20  F7                    BRA     QWINEX
```

```
                                        .SBTTL  GETIT---DOES A GET VALUE FROM THE STACK
                                        .GLOBL  TYPARG,CLRARG
011C  30            GETIT:  TSX
011D  08                    INX
011E  DF    000             STX     R0,0
0120  BD    0000G           JSR     CLRARG          ; CLR R4 THEN DO TYPARG
0123  BD    0000G           JSR     TYPARG
0126  40                    TST R0
0127  27    04              BEQ     1$
0129  86    000             LDR A   ERDOMN,I
012B  97    000             STA R0  ERRCD,0
012D  39            1$:     RTS
```

```
                                        .SBTTL VIEW----CALCULATE VIEWT PARAMETERS
                                        ; STACK AT ENTRY YMAXW, YMINW, XMAXW, XMIX
                                        ;WILL CHANGE XMINS,DLTXS,YMINS,DLTYS,TEMPX

012E                            QVIEW:  SAVRTN
0134    8D      E6                      BSR     GETIT
0136    26      39                      BNE     1$
0138    8D      0000G                   JSR     DOFP
0138                                    SET     XMINW,XNEW
0141                                    SET     YMINW,YNEW
0147    00G                             .BYTE   FLEX
0148    0000G                           .WORD   TEMPX
014A    00G                             .BYTE   FLEX
014B    0000G                           .WORD   YMINS
                                        ;DLTYS=YMAXW-YMINW
014D    00G                             .BYTE   FHEX
014E    0000G                           .WORD   TEMPX
0150    A0G                             .BYTE   FHEX+FS
0151    0000G                           .WORD   YMINS
0153    00G                             .BYTE   FLEX
0154    0000G                           .WORD   DLTYS
0156    00G                             .BYTE   FRET
0157    8D      C3                      BSR     GETIT
0159    26      16                      BNE     1$
015B    8D      0000G                   JSR     DOFP
015E    00G                             .BYTE   FLEX
015F    0000G                           .WORD   TEMPX
0161    00G                             .BYTE   FLEX
0162    0000G                           .WORD   XMINS
                                        ;DLTXS=XMAXW-XMINW
0164    00G                             .BYTE   FHEX
0165    0000G                           .WORD   TEMPX
0167    A0G                             .BYTE   FHEX+FS
0168    0000G                           .WORD   XMINS
016A    00G                             .BYTE   FLEX
016B    0000G                           .WORD   DLTXS
016D    00G                             .BYTE   FRET
016E    8D      0088'                   JSR     SCALEF
0171    7E      0000G           1$:     JMP     DREXTA
```

```
                                        .SBTTL  SCALE   SETUP SCALE FACTOR CALCULATIONS
                                        ;ROUTINE TO CALCULATE X & Y AXIS SCALE FACTORS
                                        ; STACK OF ENTRY Y, X SCALE (USER UNIT/GDU)
                    0001                SCL     = 1
                                        .GLOBL  R20
                                        .GLOBL  PSHRET,RTRN

                                        .IF     DF,SCL
0174    BD      0000G           QSCALE: JSR     PSHRET
0177    8D      03                      BSR     GETIT           ; GET THE TWO ARGUMENTS
0179    27      03                      BEQ     1$
017B    7E      0000G                   JMP     RTRN
017E    CE      0000G           1$:     LDX     YMINWI          ; Y AXIS
0181    DF      00G                     STX     R20,D           ; POINTER TO Y IN INDEX REG.
0183    8D      11                      BSR     AXISCA
0185    CE      0000G                   LDX     XMINWI          ; X AXIS
0188    DF      00G                     STX     R20,D           ; POINTER TO X IN INDEX REG.
018A    8D      0A                      BSR     AXISCA
018C    7E      0000G                   JMP     RTRN
```

```
                                        .SBTTL  AXISCA  CALCULATE AXIS SCALE FACTORS
                                        ;SUBROUTINE TO CALCULATE AXIS SCALE FACTOR
                                        ;  INDEX REG POINT TO THE X OR Y MIN
                                        ;  STACK HAS AXIS SCALE FACTOR (USER UNITS/GDU)
                                        ;WILL CHANGE (X = X OR Y)
                                        ;       %SF, %MIN, %MAX

                                        .GLOBL  FPADD,FPSUB,FPDIV
                                        ;
018F    86      00G     AXIERR: LDA A   ERDOMN,I                ; DOMAIN ERROR
0191    97      00G             STA A   ERRCD,D
0193    7E      0000G           JMP     RTRN
                                        ;
0196    BD      0000G   AXISCA: JSR     PSHRET
0199    CE      0000G           LDX     TEMPX,I
019C    BD      0000G           JSR     PULFPN  ;TEMP HAS AXIS SCALE FACTOR (USER UNITS/GDU)

                                        ; AXIS LAST IN GDU = AXIS LAST IN USER UNITS - XMINW /AXIS SF

019F    EE      00              LDX     0,X             ; TEST FOR ARGS (= 0
01A1    2B      EC              BMI     AXIERR
01A3    27      EA              BEQ     AXIERR
01A5    86      48              LDA A   9.*8.,I
01A7    8D      5A              BSR     ADDARG
01A9    BD      0000G           JSR     PSHFPN          ;PUSH AXIS LAST
01AC    CE      0000G           LDX     FPZERO,I        ; UPDATE LAST POINT REG. (SET TO ZERO)
01AF    BD      0000G           JSR     PSHFPN
01B2    DE      00G             LDX     R21,D
01B4    BD      0000G           JSR     PULFPN
01B7    DE      00G             LDX     R20,D
01B9    BD      0000G           JSR     PSHFPN
01BC    BD      0000G           JSR     FPSUB
01BF    86      40              LDA A   8.*8.,I
01C1    8D      40              BSR     ADDARG
01C3    BD      0000G           JSR     PSHFPN          ;PUSH AXIS *SF
01C6    BD      0000G           JSR     FPDIV           ;(AXIS LAST - XMINW)/(AXIS SCALE FACTOR)
01C9    CE      0000G           LDX     TEMPX,I ;STORE NEW AXIS SF
01CC    BD      0000G           JSR     PSHFPN
01CF    DE      00G             LDX     R21,D
01D1    BD      0000G           JSR     PULFPN          ;INTO AXIS SF

                                        ; NEW AXIS MINIMUM
                                        ; -(AXIS LAST GDU)*(AXIS SF)

01D4    30                      TSX                     ; NEGATE NUMBER ON STACK
01D5    A6      01              LDA A   1,X
01D7    88      80              EOR A   128.,I
]]]]]
01D9    A7      01              STA A   1,X
01DB    8D      0000G           JSR     DOFP
01DE    C0G                     .BYTE   FHEX+FM
01DF    0000G                   .WORD   TEMPX
01E1    16                      .BYTE   FDUP            ;MIN WINDOW
01E2    00G                     .BYTE   FLIN
01E3    0000G                   .WORD   R20
01E5    00G                     .BYTE   FRET
                                        ;SET UP NEW MAX.
```

```
01E6   86  20               LDA A  4*8.,!           ;DELTA AXIS
01E8   8D  19               BSR    ADDARO
01EA   BD  0000G            JSR    PSHFPN  ;DELTA AXIS
01ED   CE  0000G            LDX    TEMPX,I
01F0   BD  0000G            JSR    PSHFPN           ; AXIS SF
01F3   BD  0000G            JSR    FPMUL            ;(DELTA AXIS)*(AXIS SF) --> DELTA IN USER UNITS
01F6   BD  0000G            JSR    FPADD            ;(  ) + AXIS MIN ---> WIN MAX IN USER UNITS
01F9   86  08               LDA A  1*8.,!
01FB   8D  06               BSR    ADDARO
01FD   BD  0000G            JSR    PULFPN           ; RESULT STORED IN AXIS MAX
0200   7E  0000G   1$:      JMP    RTRN



                            ;ROUTINE WITH DUAL ENTRY POINTS
                            ;ENTRY HERE WILL ADD A ACCUM TO R20
                            ;AND PUT RESULT IN THE IX
0203   DE  00G     ADDARO:  LDX    R20,D

                            ;SECOND ENTRY POINT
                            ;ENTRY HERE WILL ADD A ACCUM TO IX
                            ;AND PUT RESULT IN THE IX REGISTER
                            ;WILL CHANGE CONDITION CODES AND IX REGISTER
0205   DF  00G     ADDAIX:  STX    R21,D            ;STORE IX IN TEMP
0207   9B  01G              ADD A  R21+1,D          ;ADD A ACCUM TO IT
0209   97  01G              STA A  R21+1,D
020B   24  03               BCC    1$
020D   7C  0000G            INC    R21              ;INCREMENT UPPER BYTE IF CARRY FROM THE ADD
0210   DE  00G     1$:      LDX    R21,D            ;RETURN TEMP TO THE IX
0212   39                   RTS
                            .ENDC
```

```
                                .SBTTL  GRFCTL--GRAPHIC CONTROLLER
                        ;THESE SHORT ROUTINES ARE CALLED FROM EVAL AND PROCEED TO SET UP
                        ;CONSTANTS FOR THE GRAPHIC PROCESSOR
                        ;
                                .MACRO  ENTRY   ARG
                                .GLOBL  Q'ARG.,Z'ARG.
                        Q'ARG.: LDX     Z'ARG.,I
                                STX     OPRAND.D
                                BRA     GRAPH
                                .ENDM

                        0213            ENTRY   DRAW
                        021A            ENTRY   RDRAW
                        0221            ENTRY   MOVE
                        0228            ENTRY   RMOVE
                        022F            ENTRY   GIN
```

```
                                        SBTTL  GRAPH---MAIN GRAPHIC CONTROLLER FOR MOVE AND DRAW
                                ;THIS ROUTINE DOES THE NECESSARY ROUTINE FOR MAT AS WELL AS
                                ;SCALER MOVE, DRAW, RMOVE, RDRAW
                                ;AND USES TYPARG TO DO SO.
                                ;
                                        .GLOBL  GRAPH
                                        .GLOBL  FPMUL
                                        .GLOBL  ADRDEV
                                        .GLOBL  STKBLD
                                        .GLOBL  BFRALC
                                        .GLOBL  SNDBER
                                        .GLOBL  UNADR
                                        .GLOBL  IOCLNR
                                        .GLOBL  A8X
                                        .GLOBL  A5X
                                ;
0236  86  000G                  GRAPH:  LDA A   EOLTG,I         ; TAG IT SO CAN FIND IT LATER
0238  36                                PSH A
0239  BD  0000G                         JSR     ADRDEV
023C  BD  0000G                         JSR     BFRALC
023F  30                                TSX
0240  08                               INX
0241  08                               INX
0242  DF  00G                          STX     R0,D
0244  96  00G                          LDA A   IOFUNC,D                ; GIN NEEDS SPECAIL SET UP
0246  81  18                           CMP A   24.,I           ; GIN?
0248  26  03                           BNE     1$
024A  7E  02EB'                       JMP     GINSET          ; GO SET UP GIN CONTROL
024D  BD  0000G                 1$:   JSR     STKBLD          ; BUILD PROPER ORDER STACK
                                ;INFORMATION BLOCK:
                                ;       TABPTR(---X START
                                ;       LENGTH(---X END
                                ;       COLCNT(---Y START
                                ;       TCOL  (---Y END
                                ;NOTE: END IS POINTER TO LAST VALUE + 1 VALUE ENTRY
                                ;
0250  BD  0350'                         JSR     GRFSCN          ; GET X AXIS VALUES
0253  DE  00G                          LDX     R0,D            ; SET UP X AXIS VALUE POINTERS
0255  DF  00G                          STX     TABPTR,D
0257  DE  00G                          LDX     R1,D
0259  DF  00G                          STX     LENGTH,D
025B  BD  0350'                         JSR     GRFSCN          ; GO GET Y AXIS INFO
025E  DE  00G                          LDX     R0,D            ; SET UP Y AXIS STUFF
0260  DF  00G                          STX     COLCNT,D
0262  DE  00G                          LDX     R1,D
0264  DF  00G                          STX     TCOL,D
0266  96  00G                          LDA A   ERRCD,D
0268  26  34                           BNE     GRFEXT
026A  CE  02AB'                 GRFAGN: LDX     GRFNRM,I        ; DISPATCH TO RAPHIC ROUTINES
026D  96  00G                          LDA A   IOFUNC,D
026F  81  18                           CMP A   24.,I           ; TEST IF DOING GIN
0271  27  40                           BEQ     GRFGIN
0273  25  05                           BCS     1$
0275  84  3F                           AND A   77,I            ; FOR RELATIVE PLOTS
0277  CE  02AF'                       LDX     GRFREL,I
027A  80  14                    1$:   SUB A   20.,I
027C  27  02                           BEQ     2$
```

```
        027E    08                        INX
        027F    08                        INX
        0280    EE      00        2$:     LDX     0,X             ; GOT DO IT
        0282    AD      00                JSR     0,X
        0284    96      00G       GRFED:  LDA A   ERRCD,D ; ANY ERRORS?
        0286    26      16                BNE     GRFEXT
        0288    DE      00G               LDX     TABPTR,D        ; MOVE VALUE POINTERS UP ONE VALUE
        028A    BD      0000G             JSR     A8X
        028D    DF      00G               STX     TABPTR,D
        028F    9C      00G               CPX     LENGTH,D        ; TEST IF DONE
        0291    27      08                BEQ     GRFEXT
        0293    DE      00G               LDX     COLCNT,D
        0295    BD      0000G             JSR     A8X
        0298    DF      00G               STX     COLCNT,D
        029A    9C      00G               CPX     TCOL,D
        029C    26      CC                BNE     GRFAGN          ; GO FOR MORE
        029E    96      00G       GRFEXT: LDA A   A.STAT,D        ; OUTPUT OPPERATION
        02A0    2B      03                BMI     1$
        02A2    BD      0000G             JSR     SNDBFR
        02A5    BD      0000G     1$:     JSR     UNADR
        02A8    BD      0000G             JSR     IOCLNR          ; CLEAN UP STACK AND RETURN
                                  ;*********************************************************************
                                  ;THE TABLES
        02AB    0457'             GRFNRM: GRFDRW
        02AD    038C'                     GRFMOV
        02AF    03E2'             GRFREL: RELDRW
        02B1    0415'                     RELMOV
```

```
                                        .SBTTL  GIN-----THE GIN DRIVER
                                ;THIS ROUTINE REQUESTS GDU INFORMATION FROM THE DEVICE AND
                                ;PROCEEDS TO CONVERT IT TO USER UNITS
                                ;
                                        .GLOBL  INPVAL,FPA,FPB
                                        .GLOBL  MATSIZ
                                ;
                                ;
02B3  CE  0000G   GRFGIN: LDX   FPA,I           ; PUT X GDU  DATA IN FPA
02B6  DF    00G           STX   POINT,D
02B8  BD  0000G           JSR   INPVAL
02BB  CE  0000G           LDX   FPB,I           ; PUT Y GDU DATA IN FP8
02BE  DF    00G           STX   POINT,D
02C0  BD  0000G           JSR   INPVAL
02C3  96    00G           LDA A ERRCD,D
02C5  26    D7            BNE   GRFEXT
                                ;
                                ;
                                ;       XNEW=(GDU'S-VIEWPORTXMIN)*XSF+WINDOWXMIN
                                ;       XSF=(DELTAWINDOW)/(DELTAVIEWPORT)
                                ;
02C7  BD  0000G           JSR   DOFP            ; RECOVER GDU'S AND CONVERT THEM
02CA    00G             .BYTE   1,EX
02CB  0000G             .WORD   FPB
02CD    A0G             .BYTE   FHEX+FS
02CE  0000G             .WORD   YMINS
02D0    C0G             .BYTE   FHEX+FM
02D1  0000G             .WORD   YSF
02D3    80G             .BYTE   FHEX+FA
02D4  0000G             .WORD   YMINW
02D6    00G             .BYTE   FLIN
02D7  0000G             .WORD   COLCNT
02D9    00G             .BYTE   FHEX
02DA  0000G             .WORD   FPA
02DC    A0G             .BYTE   FHEX+FS
02DD  0000G             .WORD   XMINS
02DF    C0G             .BYTE   FHEX+FM
02E0  0000G             .WORD   XSF
02E2    80G             .BYTE   FHEX+FA
02E3  0000G             .WORD   XMINW
02E5    00G             .BYTE   FLIN
02E6  0000G             .WORD   TABPTR
02E8    00G             .BYTE   FRET
02E9    20    59        BRA     GRFED
```

```
                                .SBTTL  GINSET--SET UP FOR GIN INPUT
                        ;THIS SECTION SETS UP THE PROPER POINTERS FOR GIN INPUT
                        ;
02E8  DF  00G   GINSET: STX     R1,D            ; SAVE FOR AWHILE
02EA  8D  1A            BSR     GINSTK          ; EXTRACT POINTERS
02EC  DE  00G           LDX     R20,D           ; PUT THEM WHERE THEY WILL DO SOME GOOD
02EE  DF  00G           STX     COLCNT,D
02F0  DE  00G           LDX     R21,D
02F2  DF  00G           STX     TCOL,D
02F4  DE  00G           LDX     R1,D            ; GO FOR SECOND SET
02F6  8D  0000G         JSR     A9X
02F9  8D  08            BSR     GINSTK
02FB  DE  00G           LDX     R20,D           ; MOVE THESE TO PROPER AREA
02FD  DF  00G           STX     TABPTR,D
02FF  DE  00G           LDX     R21,D
0301  DF  00G           STX     LENGTH,D
0303  7E  026A'         JMP     GRFAGN          ; GO DO GIN
```

```
                                        .SBTTL  GINSTK--SETS POINTERS FOR ONE AXIS
                                ; THIS ROUTINE SETS UP R20 AND R21 AS ARRAY BOUNDS POINTERS
                                ; FOR GIN.
                                ; ONE AXIS AT A TIME.
                                ;
0309                            GINSTK: SAVR'N
030F  DF    00G                         STX     R0,D            ; FOR MATSIZ
0311  A6    01                          LDA A   1,X             ; GET TAG
0313  81    00G                         CMP A   PAETG,I         ; ARRAY ELEMENT
0315  26    04                          BNE     1$
0317  EE    04                          LDX     4,X             ; GET POINTER TO VALUE
0319  20    28                          BRA     GSTKEQ
0318  EE    02                  1$:     LDX     2,X             ; MUST BE NAMETABLE
031D  E6    04                          LDA B   NTATTR,X        ; SEE IF SIMPLE VALUE
031F  C5    40                          BIT B   SCALER,I
0321  26    13                          BNE     2$
0323  DF    00G                         STX     R21,D
0325  EE    08                          LDX     NTAPTR,X        ; GET POINTER TO FIRST VALUE
0327  BD    0CC0G                       JSR     A5X
0329  DF    00G                         STX     R20,D
032C  CE    00G                         LDX     R21,D
032E  BD    0000G                       JSR     MATSIZ          ; GET ENDPOINT
0331  BD    0000G                       JSR     A5X
0334  20    15                          BRA     GSTKEX
0336  07                        2$:     TPA                     ; MARK VALUE DEFINED
0337                                    SEI
0337  01    0F                  .BYTE   01,17
0339  C4    7F                          AND B   -1-UNDEF,I
033B  E7    04                          STA B   NTATTR,X
0330  F6    0040                        LDA B   VALUND          ; SET UNDEFINED BIT IN VALUE
0340  E7    05                          STA B   NTVAL,X
0342  06                                TAP                     ; RESTORE INTERUPTS
0343  BD    0000G                       JSR     A5X             ; SET UP POINTER TO VALUE
0346  DF    00G                 GSTKEQ: STX     R20,D
0348  BD    0000G                       JSR     A8X             ; SET END POINTER
034B  DF    00G                 GSTKEX: STX     R21,D
034D  7E    0000G                       JMP     DREXTA          ; RETURN
```

```
                                        SBTTL  GRFSCN--GRAPH STACK SCANNER
                                ;THIS ROUTINE SETS UP POINTERS TO DATA VALUES
                                ;       R0-----IS POINTER TO FIRST VALUE
                                ;       R1-----IS POINTER TO LAST VALUE+1 VALUE
                                ;       R2-----IS DESTROYED
                                ;
0350                            GRFSCN: SAVRTN                  ; SAVE RETURN INFO
0350  32                        PUL A
0351  97  01G                   STA A   DREXTA+1,D
0353  32                        PUL A
0354  97  02G                   STA A   DREXTA+2,D
0356  31                        INS                             ; THRO OUT TAG
0357  32                        PUL A                           ; GET CODE RETURNED BY TYPARG
0358  30                        TSX
0359  EE  00                    LDX     0,X                     ; GET POINTER TO REAL DATA POINTERS
035B  31                        INS
035C  31                        INS
035D  4D                        TST A                           ; TEST CODE RETURNED BY TYPARG
035E  26  08                    BNE     1$
0360  08                        INX
0361  08                        INX
0362  DF  00G                   STX     R0,D                    ; SET UP START
0364  BD  0000G                 JSR     A8X
0367  DF  00G                   STX     R1,D                    ; MARK END
0369  20  1E                    BRA     GRAFRT                  ; EXIT
0368  81  10          1$:     CMP A   AMAT,I                  ; IS IT A MATRIX?
036D  26  16                    BNE     GRAFER                  ; ERROR EXIT
036F  DF  00G                   STX     R2,D                    ; TEMP. SAVE
0371  EE  02                    LDX     2,X                     ; GET POINTER TO NAMETABLE
0373  EE  0B                    LDX     NTAPTR,X                ; GET POINTER TO ARRAY
0375  BD  0000G                 JSR     A5X
0378  DF  00G                   STX     R0,D                    ; POINTER TO FIRST VALUE
037A  DE  00G                   LDX     R2,D
037C  EE  04                    LDX     4,X
037E  BD  0000G                 JSR     A5X
0381  DF  00G                   STX     R1,D                    ; POINTER TO LAST VALUE
0383  20  04                    BRA     GRAFRT                  ; EXIT
0385  86  00G           GRAFER: LDA A   ERUNDF,I
0387  97  00G                   STA A   ERRCD,D                 ; SET ERROR
0389  7E  0000G         GRAFRT: JMP     DREXTA                  ; REUTNR
```

```
                                          .SBTTL -----------------------------------------------
                                          .SBTTL ROUTINES TO MOVE, RMOVE AND RDRAW


                                          ;*****************************************************************
                                          ;
                                          ;                 ROUTINES TO RDRAW, RMOVE AND MOVE
                                          ;
                                          ;*****************************************************************



                                          .SBTTL MOVE SUBROUTINE TO MOVE
                                          ; GET X & Y VALUES, AND STORE IN XNEW YNEW & XLAST YLAST

                                          ;STACK AT ENTRY YVALUE,XVALUE

        038C    BD     0000G      GRFMOV: JSR    DOFP     ; SET XLAST AND XNEW
        038F    00G                       .BYTE  FHIN
        0391    0000G                     .WORD  TABPTR
        0392    16                        .BYTE  FDUP
        0393    00G                       .BYTE  FLEX
        0394    0000G                     .WORD  XNEW
        0396    00G                       .BYTE  FLEX
        0397    0000G                     .WORD  XLAST
        0399    00G                       .BYTE  FHIN            ; SET YLAST AND YNEW
        039A    0000G                     .WORD  COLCNT
        039C    16                        .BYTE  FDUP
        039D    00G                       .BYTE  FLEX
        039E    0000G                     .WORD  YNEW
        03A0    00G                       .BYTE  FLEX
        03A1    0000G                     .WORD  YLAST
        03A3    00G                       .BYTE  FRET
        03A4    BD     03A8'              JSR    PLOTVL
        03A7    39                        RTS
                                          .SBTTL PLOTVL PLOT A VECTOR LINE
                                          ;RELATIVE DRAW
                                          ; XNEW CONTAINS POSITION FOR X AXIS IN USER UNITS (VIRTUAL SPACE)
                                          ; YNEW CONTAINS POSITION FOR Y AXIS IN USER UNITS

                                          .GLOBL  PRIVGI          ; PRINT ROUTINE
                                          .GLOBL  CRLF
                                          .GLOBL  FPA

                                          ; Y IN GDU'S = (YNEW-YMINW)/YSF + YMINS

        03A8                      PLOTVL: ; X IN GDU'S = (XNEW-XMINW)/YSF +XMINS
        03A8    BD     0000G              JSR    DOFP
        03AB    00G                       .BYTE  FHEX
        03AC    0000G                     .WORD  XNEW
        03AE    A0G                       .BYTE  FHEX+FS
        03AF    0000G                     .WORD  XMINW
        03B1    E0G                       .BYTE  FHEX+FD
        03B2    0000G                     .WORD  XSF
        03B4    80G                       .BYTE  FHEX+FA
        03B5    0000G                     .WORD  XMINS
        03B7    00G                       .BYTE  FSAV            ; SAVE IT
```

```
0388    00G                     .BYTE   FRET
0389    8D      17              BSR     PLTIT           ; PLOT IT ONTO THE I/O BUS
038B    BD      0000G           JSR     DOFP
038E    00G                     .BYTE   FHEX
039F    0000G                   .WORD   YNEW
03C1    A0G                     .BYTE   FHEX+FS
03C2    0000G                   .WORD   YMINW
03C4    E0G                     .BYTE   FHEX+FD
03C5    0000G                   .WORD   YSF
03C7    80G                     .BYTE   FHEX+FA
03C8    0000G                   .WORD   YMINS
03CA    00G                     .BYTE   FSAV            ; SAVE IT SO OUTPUT ROUTINE CAN FIND IT
03CB    00G                     .BYTE   FRET
03CC    8D      04              BSR     PLTIT

03CE    BD      0000G           JSR     CRLF            ; APPEND A "CR"
03D1    39                      RTS                     ; RETURN
                        ;*****************************************************************
                        ;       PLOT ON TO BUS ROUTINE
03D2    7F      0000G   PLTIT:  CLR     ERRCD
03D5    CE      0000G           LDX     FPA,I           ; RECAL POINTER TO VALUE
03D8    DF      00G             STX     POINT,D         ; PUT IT WHERE I/O PROCESSOR CAN FIND IT
03DA    86      02              LDA A   2,I             ; SET COMPRESSED FORMAT FLAG
03DC    97      00G             STA A   IOFLGS,D
03DE    BD      0000G           JSR     PR!VAL          ; PRINT IT
03E1    39                      RTS
```

```
                                        .SBTTL RDRAW RELATIVE DRAW SUBROUTINE
                                        ; DY, DX IN USERS UNITS ON STACK

03E2  BD  042C'   RELDRW: JSR   RPLOT          ; GET VALUE IN VIRTUAL SPACE
03E5  BD  0000G           JSR   DOFP
03E8                      SET   XNEW,FPB
03E8    00G               .BYTE FHEX
03E9  0000G               .WORD XNEW
03EB    00G               .BYTE FLEX
03EC  0000G               .WORD FPB
03EE                      SET   YNEW,FPC
03EE    00G               .BYTE FHEX
03EF  0000G               .WORD YNEW
03F1    00G               .BYTE FLEX
03F2  0000G               .WORD FPC
03F4                      SET   XNEW,TEMPX
03F4    00G               .BYTE FHEX
03F5  0000G               .WORD XNEW
03F7    00G               .BYTE FLEX
03F8  0000G               .WORD TEMPX
03FA                      SET   YNEW,TEMPY
03FA    00G               .BYTE FHEX
03FB  0000G               .WORD YNEW
03FD    00G               .BYTE FLEX
03FE  0000G               .WORD TEMPY
0400  00G                 .BYTE FRET
0401  BD  0467'           JSR   GRFDR.         ; GO DRAW AND CLIP
0404  BD  0000G           JSR   DOFP           ; SET UP XLAST,YLAST
0407                      SET   FPB,XLAST
0407    00G               .BYTE FHEX
0408  0000G               .WORD FPB
040A    00G               .BYTE FLEX
040B  0000G               .WORD XLAST
040D                      SET   FPC,YLAST
040D    00G               .BYTE FHEX
040E  0000G               .WORD FPC
0410    00G               .BYTE FLEX
0411  0000G               .WORD YLAST
0413  00G                 .BYTE FRET
0414  39                  RTS

                                        .SBTTL RMOVE RELATIVE MOVE SUBROUTINE
                                        ;RELATIVE MOVE
                                        ; DY, DX IN USER UNITS ON STACK

0415  BD  042C'   RELMOV: JSR   RPLOT          ; DO IT
0418  BD  0000G           JSR   DOFP
041B                      SET   XNEW,XLAST
041B    00G               .BYTE FHEX
041C  0000G               .WORD XNEW
041E    00G               .BYTE FLEX
041F  0000G               .WORD XLAST
0421                      SET   YNEW,YLAST
0421    00G               .BYTE FHEX
]]]]]
0422  0000G               .WORD YNEW
0424    00G               .BYTE FLEX
0425  0000G               .WORD YLAST
```

RMOVE RELATIVE MOVE SUBROUTINE

```
26          0427    006                     .BYTE   FRET
27          0428    BD      0398'           JSR     PLOTVL
28          0428    39                      RTS
29
```