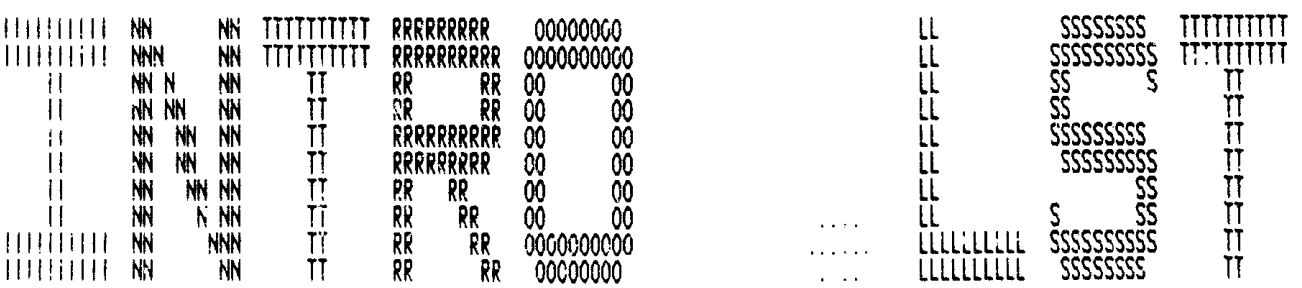


14-OCT-76

```
**********************************************************************
**********************  NOTE !!!!!!  *********************************
**********************************************************************

      THERE ARE TWO CONSTANT AND GLOBAL DECLARATION FILES
WHICH WERE USED IN MANY OF THE I/O AND EVALUATOR MODULES.
THESE FILES SERVED AS PREAMBLES TO THOSE MODULES AND WERE NOT
LISTED WITH THOSE MODULES IN AN EFFORT TO CONSERVE PAPER.
FOR YOUR CONVENIENCE THE TWO PREAMLE FILES ARE LISTED HERE AS:

      EQU  .MAC ---- THE EVALUATOR PREAMBLE.

      IOEQU .MAC ---- THE I/O SYSTEM PREAMBLE.

IF YOU ARE READING AN EVALUATOR OR AN I/O SYSTEM
MODULE IT MAY BE HELPFUL TO HAVE THESE LISTINGS
HANDY.  ALSO, FURTHER DEFINITIONS OF MANY OF THE VARIABLES
WILL BE FOUND IN THE FORMAL DOCUMENTATION LISTING.

      THE PREAMBLE FILES WILL FOLLOW.
```

```
*****************************************************************
**********************.EQU  .MAC **********************************
*****************************************************************
******************** EVALUATOR PREAMBLE ***************************
*****************************************************************
;
;     GENEERAL PURPOSE MACROS
;
      .MACRO  PLRA          ;PULL RTRN ADDR & SAVE IT IN DREXTA
      PUL A
      STA A   DREXTA+1,D
      PUL A
      STA A   DREXTA+2,D
      .ENDM
;
      .MACRO  PLRB          ;PULL RTRN ADDR & SAVE IT IN DREXTB
      PUL A
      STA A   DREXTB+1,D
      PUL A
      STA A   DREXTB+2,D
      .ENDM
;
;     EVALUATOR CONTROL CONSTANTS
;
      .GLOBL  ASGCOD        ;ASSIGN OPERATOR
      .GLOBL  EQUCOD        ;FIRST RELATIONAL OPERATOR
      .GLOBL  FNACOD        ;FIRST USER FUNCTION TOKEN
      .GLOBL  SIZCOD        ;FIRST ON UNIT CODE
      .GLOBL  MULCOD        ;MULTIPLY CODE
      .GLOBL  EOFCOD        ;EOF TOKEN - EOF ON UNITS
      .GLOBL  MPLCOD        ;UNARY PLUS OPERATION
      .GLOBL  MMICOD        ;UNARY MINUS OP
      .GLOBL  CRCOD         ;END OF LINE
      .GLOBL  CONCOD        ;CONSTANT
      .GLOBL  LITCOD        ;LITERAL
      .GLOBL  DATCOD        ;DATA STATEMENT
      .GLOBL  IMACOD        ;IMAGE STATEMENT
      .GLOBL  LSTCOD        ;LAST VALID TOKEN
```

```
;
;       CONSTANTS IN PAGE ZERO
;
        .GLOBL  CLPTR           ;CURRENT LINE POINTER
        .GLOBL  NLPTR           ;NEXT LINE POINTER
        .GLOBL  NTPTR           ;NEXT TOKEN POINTER
        .GLOBL  CTKN            ;CURRENT TOKEN
        .GLOBL  LCLFLG          ;LOCAL CONTROL FLAGS
EXTFLG  =  H80                  ;EXIT LINE CONTROL GROUP
ABRFLG  =  H40                  ;ABORT CURRENT LINE
IMXFLG  =  H20                  ;THIS IS IMMEDIATE EXECUTE LINE
FNFLG =  H10                    ;PROCESSING USER DEFINED FUNCTION
DIMFLG  =  H04                  ;DIMENSION STATEMENT ACTIVE
ONSFLG  =  H02                  ;ON STATEMENT IN DEFINITION MODE
;
        .GLOBL  GLBFLG          ;GLOBAL CONTROL FLAGS
RUNFLG  =  H80                  ;RUN MODE
STPFLG  =  H40                  ;STEP MODE
TRCFLG  =  H20                  ;TRACE MODE
KEYFLG  =  H10                  ;FUNCTION KEYS DISABLED
DISSRQ  =  H08                  ;DISABLE SRQ INTERRUPTS
QUAFLG  =  H02                  ;TSTINT ACTIVATED ON UNIT IN LAST CALL
NCSFLG  =  H01                  ;CASE DIFFERANCE TEST
        .GLOBL  OPRADR          ;OPERATOR ADDRESS SAVE AREA
        .GLOBL  DREXTA          ;DIRECT EXIT ADDRESS A
        .GLOBL  DREXTB          ;DIRECT EXIT ADDRESS B
;
        .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7 ;PSEUDO REGISTERS
        .GLOBL  R8,R9,R10,R11
;
        .GLOBL  ERRCD           ;ERROR CODE, NO ERROR=0
        .GLOBL  ERRCDB          ;BACK UP AREA FOR SIZE ERROR HOLDING
        .GLOBL  PNDFLG          ;INTERRUPTS PENDING FLAGS
;
;       BIT     INDEX   TYPE
;       7       0       BREAK
;       6       2       SIZE
;       5       4       PAGE FULL
;       4       6       SRQ
;       3       8       EOI
;       2       10      E11
;       1       12      E12
;       0       14      E12
;
;       -       16      EOF (10 CELLS)
;
        .GLOBL  PNDEOF          ;I/O UNIT WITH EOF CONDITION
        .GLOBL  LSP             ;HIGH END OF DATA STACK
        .GLOBL  SBP             ;HIGH END OF STACK
        .GLOBL  PGMPTR          ;POINTER TO FIRST LINE OF PGM
        .GLOBL  STPTR           ;SYMBOL TABLE POINTER
```

```
;
;       TABLES THAT NEED NOT BE IN PAGE ZERO
;
        .GLOBL  ONTBL           ;ON UNIT LINE POINTERS
        .GLOBL  EOFTBL          ;ON UNITS FOR EOF
        .GLOBL  FNTBL           ;USER FUNCTION LINE POINTERS
        .GLOBL  KEYSTK          ;FUNCTION KEY STACK - 8 SLOTS
;                               ;  9 BYTES LONG - LAST BYTE=0
;
        .GLOBL  JMPX            ;FANCY JUMP
;       STA A   JMPX+4
;       LDX     0,X
;       JMP     0,X
;
        .GLOBL  LDXX            ;FANCY LOAD
;       STA A   LDXX+4
;       LDX     0,X
;       RTS
;
        .GLOBL  LDAX            ;FANCY LOAD ACCX
;       STA A   LDAX+4
;       LDA A   0,X
;       RTS
;
        .GLOBL  LDBX            ;ANOTHER FNACY LOAD
;       STA A   LDBX+4
;       LDA B   0,X
;       RTS
;
        .GLOBL  STAX            ;FANCY STORE
;       STA A   STAX+4
;       STA B   0,X
;       RTS
;
        .GLOBL  JMPAX           ;FANCY JUMP
;       STA A   JMPAX+4
;       JMP     0,X
;
        .GLOBL  CDSPTR          ;CRNT DATA STATEMENT
        .GLOBL  CDOPTR          ;CRNT DATA OBJECT
```

```
;
;       NAME TABLE FORMAT
;
NTLINK          = 0                     ;POINTER TO NEXT ENTRY
NTNAME          = 2                     ;OBJECT NAME
;                       FLAG BITS
NTATTR          = 4                     ;ATTRIBUTE FLAGS
UNDEF = H80                     ;UNDEFINED
SCALER          = H40                   ;SCALER VALUE
ARRAY = H20                     ;ARRAY ENTRY
STRING          = H10                   ;STRING ENTRY
PARM = H08                      ;USER FUNCTION PARM
ALLOK = H04                     ;ALL ELEMENTS OF MATRIX ARE DEFINED  !
;                       VALUE ENTRY
NTVAL = 5                       ;8 BYTE FLOATING POINT NUMBER
;                       ARRAY ENTRY
NTWROW          = 5                     ;WORKING ROW SIZE
NTWCOL          = 7.                    ;COLUMN
NTDIMS          = 9.                    ;DIMENSIONED SIZE
NTAPTR          = 11.                   ;DATA POINTER
;                       STRING ENTRY
NTDLEN          = 5                     ;DIMENSIONED LENGTH
NTWLEN          = 7                     ;WORKING LENGTH
NTSPTR          = 11.                   ;DATA POINTER
;
;       DATA OBJECTS IN MEMORY
;
OBJLEN          = 0                     ;LENGTH OF ENTRY
VALERR          = H40                   ;STRING IS UNDEFINED
OBJATR          = 2                     ;ATTRIBUTE FLAGS
OBJBCK          = 3                     ;BACK POINTER TO NAME TABLE
OBJDT = 5                       ;DATA
;
;       PROGRAM OBJECT LINE
;
PGMLEN          = 0                     ;LENGTH OF THIS ENTRY
PGMATR          = 2                     ;ATTRIBUTE BYTE
PGMFP = 3                       ;FOWARD POINTER
PGMBP = 5                       ;BACK POINTER
PGMLNN          = 7                     ;LINE NUMBER
PGMCD = 9                       ;OBJECT CODE ORIGIN
```

```
;
;       STACK TAGS
;
        .GLOBL  NULLTG          ;NULL TAG - DELETED ENTRY
        .GLOBL  PLOSTG          ;POINTER TO LITERAL IN OBJECT STRING
        .GLOBL  ESTG            ;EVALUATOR STATUS
        .GLOBL  GOSTG           ;GOSUB/RETURN ENTRY
        .GLOBL  FORTG           ;FOR/NEXT ENTRY
        .GLOBL  LISTTG          ;LIST COUNT FOR LIST COMMAND
        .GLOBL  PGMTG           ;PROGRAM POINTER
        .GLOBL  IMXTG           ;IMMEDIATE EXECUTE LINE POINTER
        .GLOBL  PNTSTG          ;POINTER TO NAME TABLE - STRING
        .GLOBL  PSCTG           ;POINTER TO STRING COUNT
        .GLOBL  PNTNTG          ;POINTER TO NAME TABLE - NUMERIC
        .GLOBL  PAETG           ;POINTER TO ARRAY ELEMENT
        .GLOBL  VALTG           ;F. P. VALUE
        .GLOBL  LNNOTG          ;LINE NUMBER
        .GLOBL  ITM1TG          ;2 BYTE ITEM TYPE 1
        .GLOBL  ITM2TG          ;2 BYTE ITEM TYPE 2
        .GLOBL  LBRKTG          ;LEFT BRACKET FOR SUBSCRIPT AND DIMENSION
        .GLOBL  SEMITG          ;SEMICOLONFOR PRINT
        .GLOBL  ATSNTG          ;AT SIGN TAG
        .GLOBL  BRKSTG          ;STACK STUFF FOR PRINT
        .GLOBL  ALLTG           ;ALL FOR DELETE ALL
        .GLOBL  RTRNTG          ;RETURN ADDRESS
        .GLOBL  PRITG           ;PRINT ENTRY
        .GLOBL  CALLTG          ;CALL STATEMENT NAME (CALL CONVERTS IT)
        .GLOBL  EOLTG           ;END OF LINE
        .GLOBL  EOSTG           ;END OF STACK
;
;       R4 BITS FROM TYPARG AND TYPRES
;
RSTR    =  H80                  ;RESULT STRING
RMAT    =  H40                  ;RESULT MATRIX
RFAIL   =  H0C0                 ;ERROR IN RESULT
ASTR    =  H20                  ;FIRST OPERAND STRING
AMAT    =  H10                  ;FIRST OPERAND MATRIX
AFAIL   =  H30                  ;FIRST OPERAND ERROR
BSTR    =  H08                  ;SECOND OPERAND
BMAT    =  H04
CSTR    =  H02                  ;THIRD OPERAND
CMAT    =  H01
```

```
;
;       ERROR CODES
;
        .GLOBL  ERNOFN          ;NO FUNCTION - FUNCTION NOT DEFINED
        .GLOBL  ERNIMX          ;NOT IN IMMEDIATE EXEC MODE
        .GLOBL  ERWSFL          ;STACK OVERFLOW
        .GLOBL  ERDOMN          ;OPERAND IS INVALID
        .GLOBL  ERLNNF          ;LINE NOT FOUND
        .GLOBL  ERSTOP          ;STOP PROGRAM - STOP STATEMENT
        .GLOBL  EROFA           ;OF ARGUMENT IS INVALID
        .GLOBL  ERFORA          ;FOR ARGUMENT IS INVALID
        .GLOBL  ERNXTA          ;NEXT STATEMENT ARGUMENT
        .GLOBL  ERNDT           ;NO DATA STATEMENT/TOO LITTLE DATA
        .GLOBL  ERBRK           ;BREAK
        .GLOBL  ERFOFN          ;LOGICAL UNIT NO IN ON/OFF STMT IS INVALID
        .GLOBL  ERASGN          ;SYNTAX ERROR IN ASSIGN
        .GLOBL  ERUNDF          ;UNDEFINED VARIABLE
        .GLOBL  ERSHAP          ;SHAPE OF MATRIX (RANK ERROR)
        .GLOBL  ERVAL           ;VAL FUNCTION FAILED
        .GLOBL  ERNXFN          ;EXTENDED FUNCTION ROM PACK NOT THERE
;
;*********************** END OF EQU   MAC ******************************
;*************************************************************************
```

```
;*****************************************************************
;********************* IOEQU.MAC *********************************
;*****************************************************************
;***************** I/O SYSTEM PREAMBLE ***************************
;*****************************************************************
;
;
;       CONSTANTS IN PAGE ZERO
;
        .GLOBL  CLPTR           ;CURRENT LINE POINTER
        .GLOBL  NLPTR           ;NEXT LINE POINTER
        .GLOBL  NTPTR           ;NEXT TOKEN POINTER
        .GLOBL  CTKN            ;CURRENT TOKEN
        .GLOBL  LCLFLG          ;LOCAL CONTROL FLAGS
EXTFLG  =  H80                  ;EXIT LINE CONTROL GROUP
ABRFLG  =  H40                  ;ABORT CURRENT LINE
FNFLG =  H10                    ;PROCESSING USER DEFINED FUNCTION
DIMFLG  =  H04                  ;DIMENSION STATEMENT ACTIVE
ONSFLG  =  H02                  ;ON STATEMENT IN DEFINITION MODE
;
        .GLOBL  GLBFLG          ;GLOBAL CONTROL FLAGS
RUNFLG  =  H80                  ;RUN MODE
STPFLG  =  H40                  ;STEP MODE
TRCFLG  =  H20                  ;TRACE MODE
KEYFLG  =  H10                  ;FUNCTION KEYS DISABLED
DISSRQ  =  H08                  ;DISABLE SRQ INTERRUPTS
;
        .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16
        .GLOBL  R17,R18,R19,R20,R21,R22,R23
;PSEUDO REGISTERS
;
        .GLOBL  ERRCD           ;ERROR CODE, NO ERROR=0
        .GLOBL  SYSERR          ;SYSTEM ERROR ROUTINE
        .GLOBL  PNDFLG          ;INTERRUPTS PENDING FLAGS
        .GLOBL  PNDEOF          ;I/O UNIT WITH EOF CONDITION
        .GLOBL  LSP             ;HIGH END OF DATA STACK
        .GLOBL  SBP             ;HIGH END OF STACK
        .GLOBL  PGMPTR          ;POINTER TO FIRST LINE OF PGM
```

```
;
;       TABLES THAT NEED NOT BE IN PAGE ZERO
;
        .GLOBL  KEYSTK          ;FUNCTION KEY STACK - 8 SLOTS
;                               ;  9 BYTES LONG - LAST BYTE=0
;
        .GLOBL  JMPX            ;FANCY JUMP
;       STA A   JMPX+4
;       LDX     0,X
;       JMP     0,X
;
        .GLOBL  LDXX            ;FANCY LOAD
;       STA A   LDXX+4
;       LDX     0,X
;       RTS
;
        .GLOBL  LDAX            ;FANCY LOAD ACCX
;       STA A   LDAX+4
;       LDA A   0,X
;       RTS
;
        .GLOBL  CDSPTR          ;CRNT DATA STATEMENT
        .GLOBL  CDOPTR          ;CRNT DATA OBJECT
;
        .GLOBL  BRKCNT          ;FOR DIMENSION
```

```
;
;       NAME TABLE FORMAT
;
NTLINK          = 0                     ;POINTER TO NEXT ENTRY
NTNAME          = 2                     ;OBJECT NAME
;                       FLAG BITS
NTATTR          = 4                     ;ATTRIBUTE FLAGS
UNDEF = H80                     ;UNDEFINED
SCALER          = H40                   ;SCALER VALUE
ARRAY = H20                     ;ARRAY ENTRY
STRING          = H10                   ;STRING ENTRY
PARM = H08                      ;USER FUNCTION PARM
;                       VALUE ENTRY
NTVAL = 5                       ;8 BYTE FLOATING POINT NUMBER
VALIND          = H40                   ; UNDEFINED BIT IN F.P. VALUE
;                       ARRAY ENTRY
NTWROW          = 5                     ;WORKING ROW SIZE
NTWCOL          = 7.                    ;COLUMN
NTDIMS          = 9.                    ;ORIGINAL DIMENSIONED SIZE IN BYTES
NTAPTR          = 11.                   ;DATA POINTER
;                       STRING ENTRY
NTDLEN          = 5                     ;DIMENSIONED LENGTH
NTWLEN          = 7                     ;WORKING LENGTH
NTSPTR          = 11.                   ;DATA POINTER
;
;       DATA OBJECTS IN MEMORY
;
OBJLEN          = 0                     ;LENGTH OF ENTRY
OBJATR          = 2                     ;ATTRIBUTE FLAGS
OBJBCK          = 3                     ;BACK POINTER TO NAME TABLE
OBJDT = 5                       ;DATA
;
;       PROGRAM OBJECT LINE
;
PGMLEN          = 0                     ;LENGTH OF THIS ENTRY
PGMATR          = 2                     ;ATTRIBUTE BYTE
PGMFP = 3                       ;FOWARD POINTER
PGMBP = 5                       ;BACK POINTER
PGMLNN          = 7                     ;LINE NUMBER
PGMCD = 9.                      ;OBJECT CODE ORIGIN
;
;       FLOATING POINT SYSTEM CONSTANTS
;
;
FIX1A = 3               ; POSITION OF INTEGER IN NUMBER ON STACK
FIX1B = 4               ; WITH POINTER AT TAG.
```

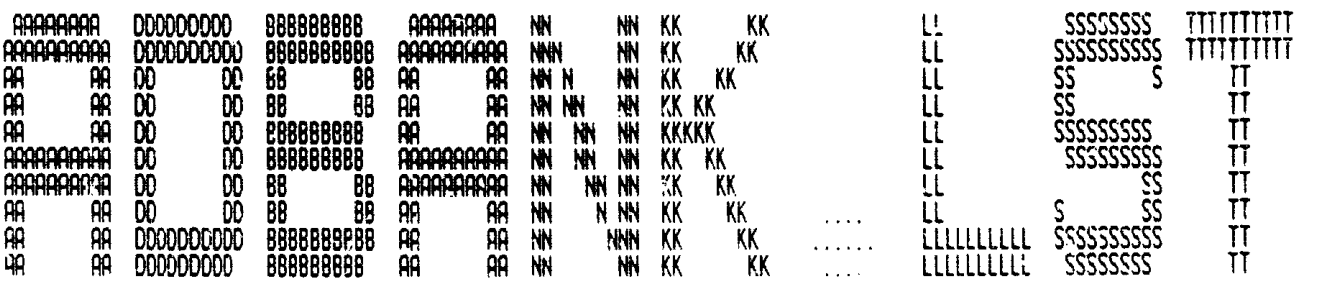
```
;
;
;       STACK TAGS
;
        .GLOBL  NULLTG          ;NULL TAG - DELETED ENTRY
        .GLOBL  PLOSTG          ;POINTER TO LITERAL IN OBJECT STRING
        .GLOBL  ESTG            ;EVALUATOR STATUS
        .GLOBL  GOSTG           ;GOSUB/RETURN ENTRY
        .GLOBL  FORTG           ;FOR/NEXT ENTRY
        .GLOBL  LISTTG          ;LIST COUNT FOR LIST COMMAND
        .GLOBL  PGMTG           ;PROGRAM POINTER
        .GLOBL  IMXTG           ;IMMEDIATE EXECUTE LINE POINTER
        .GLOBL  PNTSTG          ;POINTER TO NAME TABLE - STRING
        .GLOBL  PSCTG           ;POINTER TO STRING COUNT
        .GLOBL  PNTNTG          ;POINTER TO NAME TABLE - NUMERIC
        .GLOBL  PAETG           ;POINTER TO ARRAY ELEMENT
        .GLOBL  VALTG           ;F. P. VALUE
        .GLOBL  LNNOTG          ;LINE NUMBER
        .GLOBL  ITM1TG          ;2 BYTE ITEM TYPE 1
        .GLOBL  ITM2TG          ;2 BYTE ITEM TYPE 2
        .GLOBL  ATSNTG          ;  OR   TAG
        .GLOBL  LBRKTG
        .GLOBL  RTRNTG          ;RETURN ADDRESS
        .GLOBL  PRTTG           ;PRINT ENTRY
        .GLOBL  EOLTG           ;END OF LINE
        .GLOBL  EOSTG           ;END OF STACK
        .GLOBL  SEMITG          ; SEMI-COLON TAG
        .GLOBL  LOCTG           ; ROUTINE TO LOCATE TAGS ON THE STACK
        .GLOBL  BAKSTG          ; MARKS INFORMATION FROM STKBLD (4 BYTES)
;
;       R4 BITS FROM TYPARG AND TYPRES
;
RSTR    =  H80                  ;RESULT STRING
RMAT    =  H40                  ;RESULT MATRIX
RFAIL   =  H0C0                 ;ERROR IN RESULT
ASTR    =  H20                  ;FIRST OPERAND STRING
AMAT    =  H10                  ;FIRST OPERAND MATRIX
AFAIL   =  H30                  ;FIRST OPERAND ERROR
BSTR    =  H08                  ;SECOND OPERAND
BMAT    =  H04
CSTR    =  H02                  ;THIRD OPERAND
CMAT    =  H01
```

```
;
;   ERROR CODES
;
;
        GLOBL   ERDOMN          ; DOMAIN ERROR ON INPUT
        GLOBL   ERNSEP          ;NO SEPARATOR ON INPUT
        GLOBL   ERTERM          ;TERMINATION MESSAGE FOR BREAK 2
        GLOBL   ERFBFR          ; FULL BUFFER (INTERNAL ERROR)
        GLOBL   ERIOE           ; IOE ON IEC BUS
        GLOBL   ERFILE          ; FILE PARAMETER ERROR
        GLOBL   ERATSN          ; . ARGUMENT ERROR (ADDRESS OUT OF BOUNDS
        GLOBL   EREOM           ; END OF MEDIUM ON MAGTAPE (LOGICAL OR PHYSICAL)
        GLOBL   ERUNDF          ; UNDEFINED VARIBALE USE
        GLOBL   ERNIOD          ; ILLEGAL I/O ATTEMPTED ON DEVICE
;
```

```
;
;OTHER GOOD I/O PAGE ZERO PUMPLESTILTSKINS
;
        .GLOBL  CRSTAT          ; INPUT BUFFER EOL STATUS BYTE
                                ; (CRSTAT=0 IF NO EOL IN PRESENT BUFFER)
CRNORM  =  H01          ; NORMAL "CR" FOUND
CRDC3   =  H02          ; DC3 FOUND (  MODE)
CREOI   =  H04          ; EOI FOUND
CREOF   =  H08          ; EOF FOUND (FILE ONLY)
CRETX   =  H10          ; ETX FOUND (  MODE)
CREOT   =  H20          ; EOT FOUND (INTERNAL TAPE)
EOFTYP  =  H38          ; (EOF+ETX+EOT)
CRVLD   =  H80          ; CRSTAT IS VALID IF SET (
                        ; (REACHED END OF BUFFER)
        .GLOBL  CHAR            ;LAST CHARACTER XFERED IN I/O
        .GLOBL  IOFUNC          ;PRESENT I/O OP
        .GLOBL  A.STAT          ; CHANNEL A STATUS
        .GLOBL  A.PRIM          ; CHAN. A PRIMARY ADDRESS
        .GLOBL  A.SEC           ; CHAN. A SECONDARY ADDRES
        .GLOBL  A.PTR           ; POINTER TO CHAN. A BUFFER
        .GLOBL  A.STRT          ; START OF BUFFER
        .GLOBL  A.END           ; END OF BUFFER
        .GLOBL  A.MAX           ; MAXIMUM VALID POINTER
;
;       STATUS BYTE FORMAT (A.STAT)
DIRCT   =  H80                  ; DIRECTION 0-OUTPUT 1-INPUT
BFRSTT  =  H20                  ; BUFFER STATUS 0-EMPTY 1-FULL
BUSACT  =  H10                  ; ACTIVE BUS 0-NO 1-YES
FMTVLD  =  H08                  ; SET IF USING STATEMENT APPLIES
ATVLD   =  H04                  ; SET IF  OR   INFO IS ON STACK
SECDEF  =  H02                  ; SECONDARY DEFINED 0-NO 1-YES
PRIDEF  =  H01                  ; PRIMARY DEFINED 00-NO 1-YES
;
        .GLOBL  KBIN            ;LAST INPUT KEY
NOKEY   =  H80                  ;NOKEY CODE
        .GLOBL  KBFLAG          ;KEYBOARD CONTROL CODES
;       CONTROL BITS
RPTCTL  =  H80                  ; REPEAT CONTROL FOR FUNCTION KEYS
ENDKEY  =  H40                  ; "CR" OR EQUAL
STPKEY  =  H20                  ; STEP KEY (NOT RUN ONLY)
NTRFLY  =  H10                  ; SET IF NO INTERUPT!
ATLOD   =  H08                  ; SET IF AUTO LOAD KEY (NOT RUN ONLY)
RECLFG  =  H04                  ; RECALL K / WITH LINE NO.  IN BUFFER
RUNN    =  H02                  ; RUN MODE FOR THE KEYBOARD
INPUTK  =  H01                  ; INPUT MODE FOR THE KEYBOARD
;
        .GLOBL  DISCNT          ; IDLE LOOP COUNTER FOR BLINKING CURSOR
        .GLOBL  BLINK           ; WRITE-THRU CONTROL
        .GLOBL  CURSOR          ; THE CHARACTER TO USE FOR THE CURSOR
        .GLOBL  PPMODE          ;  MODE FLAG
        .GLOBL  NOOUT           ; OUTPUT BUFFER FLAGS
NOWRIT  =  H01                  ; IF SET NEVER OUTPUT BUFFER ( FOR UNLEX)
LSTFMT  =  H02                  ; SET TO OUTPUT IN LIST MODE
SNDIT   =  H80                  ; SET TO PUT EOI WITH LAST BYTE IN BUFFER
```

```
;
;       THINGS FOR FORMATED PRINT
;
        .GLOBL  DP              ; DATA POINTER
        .GLOBL  DL              ; DATA LENGTH
        .GLOBL  DT              ; DATA TYPE 1-VALUE,2-STRING
;
;I/O PROCESSOR FLAGS, POINTERS
;
        .GLOBL  OPRADR          ; I/O OPERATION SEED
        .GLOBL  TAGPTR          ; POINTER TO PRESENT TAG IN I/O LIST
        .GLOBL  LENGTH          ; DIMENSIONED LENGTH OF PROC. VAR.
        .GLOBL  POINT           ; POINTER TO INFO. IN MEM. TO BE XFERED
        .GLOBL  CHRCNT          ; ACTUAL XMIT. LENGTH OF CHAR. STRINGS
        .GLOBL  TABPTR          ; POINTER TO PRESENT NAME TABLE
        .GLOBL  COLCNT          ; COL. COUNT OF PRESENT MATRIX
        .GLOBL  TCOL            ; TEMPORARY COLCNT
        .GLOBL  IOFLGS          ; I/O PROCESSOR CONTROL FLAGS
;
        .GLOBL  BANK            ; PRESENT BANK ADDRESS
;
; INTERNAL PRE-DEFINED DEVICE CODES
;
FILDEV = 0.                     ; FILE DEVICE
KBDEV  = 31.                    ; KEYBOARD DEVICE CODE
DSPDEV = 32.                    ; DISPLAY    "      "
MTPDEV = 33.                    ; MAG. TAPE  "      "
DATDEV = 34.                    ; DATA STATEMENT DEVICE CODE
MTPD2  = 35.                    ; MAGTAPE SEC. DEVICE FOR TWO BUFFERS
ITTDEV = 36.                    ; INTERNAL TRANSMIT DEVICE
T51DEV = 37.                    ; TEK51 ADDRESS
;
; I/O BUFFERS
;
        .GLOBL EDTBFR           ; EDIT BUFFER
        .GLOBL IOBFR1           ; GENERAL BUFFER 1
        .GLOBL MTBFR            ; MAG. TAPE BUFFER
;
;       CONVIENENCE ROUTINES
;
        .GLOBL  DREXTA          ; RETURN A
        .GLOBL  DREXTB          ; RETURN B

************** END OF IOEQU.MAC ********************************
****************************************************************
```


AAAAAAAA DDDDDDDD BBBBBBBBB AAAAAAAA NN NN KK KK LL SSSSSSSS TTTTTTTTTT
AAAAAAAAAA DDDDDDDDD BBBBBBBBBB AAAAAAAAAA NNN NN KK KK LL SSSSSSSSS TTTTTTTTTT
AA AA DD DD BB BB AA AA NN N NN KK KK LL SS S TT
AA AA DD DD BB BB AA AA NN NN NN KK KK LL SS TT
AA AA DD DD BBBBBBBBB AA AA NN NN NN KKKKK LL SSSSSSSS TT
AAAAAAAAAA DD DD BBBBBBBBB AAAAAAAAAA NN NN NN KK KK LL SSSSSSSS TT
AAAAAAAAAA DD DD BB BB AAAAAAAAAA NN NN NN KK KK LL SS TT
AA AA DD DD BB BB AA AA NN N NN KK KK .... LL S SS TT
AA AA DDDDDDDDD BBBBBBBB AA AA NN NNN KK KK ..... LLLLLLLLLL SSSSSSSSS TT
AA AA DDDDDDDDD BBBBBBBB AA AA NN NN KK KK .... LLLLLLLLLL SSSSSSS TT

14-OCT-76

.MAIN. RT-11 MMAC VM02-10 14-OCT-76 01:27:42
TABLE OF CONTENTS

```
                                .IDENT  /MEL126/

                        ; NOTE.  IF ETOX OR LOG IS IN THE XFN BANK THEN UP MUST!!!! BE ALSO.

                                .MACRO  NAME    NM
                                .IF     DF,MAIN
                                .GLOBL  NM
                        NM:
                                .ENDC
                                .IF     DF,BANK
                                .GLOBL  Q'NM
                        Q'NM:
                                .ENDC
                                .ENDM

                                .GLOBL  PSHRET,RTRN,PSHFPN,PULFPN,ERTRNG,ERRCD,ZANS,MAXANS
                                .GLOBL  CONV1,DEGCON,GRDCON,RADCON
                                .GLOBL  CTKN, JMPAX
                                .GLOBL  FPADD,FPSUB,FPMUL,FPDIV,A8X,A9X
                                .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7,T1,T2,T3,T4
                                .GLOBL  FPA,FPB,FPC,CMPFPN,TABPNT
                                .GLOBL  FPNEG,SLN,NGTAB
                                .GLOBL  DOFP,FHEX,FHIN,FRET,FLEX,FLIN,FHIM,FART
                                .GLOBL  FSAV,FRES,FSWP,FDUP
                                .GLOBL  SIGNS,FLOAT2,NORM,NORMR,TRUTH,FPCMP
                                .GLOBL  FPZERO,FPONE,FPTWO,TABAD,PICON
                                .GLOBL  OVLN2,LOG2,OVLTN,TAN15

0000                    BANK    =       0               ;   IF DEFINED GET BANK CODE
                        ;MAIN   =       0               ;IF DEFINED GET MAIN ROM CODE

                                .SBTTL  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                .SBTTL  XXXXXX                          XXXXXX

                                .IF     DF,BANK
                                .GLOBL  ADBANK
0000'                   ADBANK  =       .
                                .SBTTL  XXXXXX  B A N K   V E R S I O N  XXXXXX
                                .TITLE  ADBANK
                                .ENDC

                                .IF     DF,MAIN
                                .GLOBL  ADMAIN
                        ADMAIN  =       .
                                .SBTTL  XXXXXX  M A I N   V E R S I O N  XXXXXX
                                .TITLE  ADMAIN
                                .ENDC

                                .SBTTL  XXXXXX                          XXXXXX
                                .SBTTL  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                .NLIST  CND

]]]]]
0080                            FA=200                  ;ADD CODE
00A0                            FS=240                  ;SUB CODE
00C0                            FM=300                  ;MULT. CODE
```

```
196             00E0                FD=340          ;DIVIDE CODE
```

```
                                        .SBTTL  ACS,ASN,ATN
                                        .GLOBL  ACSCOD,SQR,ERARC
                                        ;ARC TRIG ROUTINES

                                ;ARC SIN AND COSINE
                                ;

0000                                    NAME    ACS
0000                                    NAME    ASN
0000    BD    0000G                     JSR     PSHRET

                                ;TAKE ATN(X/SQR(1-X 2))
                                ;SCREW AROUND WITH OCTANTS LATER IF REALLY COSINE

0003    BD    0000G                     JSR     DOFP
0006    00G   C0G                       .BYTE   FDUP,FDUP+FM    ;X 2
0008    00G                             .BYTE   FHEX
0009    0000G                           .WORD   FPONE
000B    A0G   00G                       .BYTE   FSWP+FS,FRET    ;1-X 2
000D    BD    0000G                     JSR     SQR
0010    96    00G                       LDA A   ERRCD,D         ;CHECK FOR SQR OF NEG NUMBER
0012    27    07                        BEQ     1$              ;ALL OK
0014    86    00G                       LDA A   ERARC,I         ;SET TRIG ERROR
0016    97    00G                       STA A   ERRCD,D
0018    7E    0000G                     JMP     ZANS            ;RETRUN WITH 0 AS ANSWER
001B    BD    0000G             1$:     JSR     FDIV            ;X/ABOVE
001E    20    03                        BRA     ATNE

                                        ;ATN FUNCTION
                                        ;REDUCE RANGE TO X<PI/12, AND APPLY CONTINUED FRACTIONS
                                        ;
0020                                    NAME    ATN
0020    BD    0000G                     JSR     PSHRET
0023    7F    0000G             ATNE:   CLR     R1              ;ASSUME NO RANGE REDUCTION UNLESS WE FIND OUT OTHERWISE
0026    30                              TSX
0027    A6    01                        LDA A   1,X
0029    97    01G                       STA A   R1+1,D          ;REMEMBER SIGN
002B    84    07                        AND A   7,I
002D    A7    01                        STA A   1,X             ;AND TAKE ABS VALUE
002F    81    04                        CMP A   4,I             ;NO IF EXP IS NEG
0031    27    10                        BEQ     1$              ;TEST AND SEE IF X>1
0033    2B    12                        BMI     2$
0035    BD    0000G             3$:     JSR     DOFP            ;YES, TAKE ATN OF 1/X, AND SUB FROM 90 DEG
                                        ;AFTER ALL OF THE SMOKE CLEARS
0038    00G                             .BYTE   FHEX
0039    0000G                           .WORD   FPONE
003B    E0G   00G                       .BYTE   FSWP+FD,FRET    ;1/X AND RETURN TO CODE
003D    86    02                        LDA A   2,I
003F    97    00G                       STA A   R1,D            ;NOTE IN R1
]]]]]
0041    20    04                        BRA     2$
0043    A6    02                1$:     LDA A   2,X             ;CHECK REST OF EXPONENT
0045    26    FE                        BNE     3$              ;EXP NOT 0, SO X>1
0047    CE    0000G             2$:     LDX     TAN15,I         ;TANGENT OF 15 DEGREES
004A    DF    00G                       STX     TABPNT,D
004C    BD    0000G                     JSR     CMPFPN          ;COMPRE TO X
004F    2B    1A                        BMI     ATN1            ;X<
0051    7C    0000G                     INC     R1              ;NOTE RANGE REDUCE
```

```
                                ;AND TAKE X=(X*SQR(3)-1)/(X+SQR(3))
0054    BD   0000G              JSR     DOFP
0057    00G  00G                .BYTE   FDUP,FSAV
0059    C0G                     .BYTE   FHIM+FM         ;X*SQR(3)
005A    0401 0DB3   D742  SQR3: .WORD   H0401, H0DCB3, H0D742, H0C265
0060    C265
0062    A0G                     .BYTE   FHEX+FS         ;-1
0063    0000G                   .WORD   FPONE
0065    00G                     .BYTE   FHEX
0066    005A'                   .WORD   SQR3            ;PUSH SQR 3
0068    80G                     .BYTE   FRES+FA
0069    E0G  00G                .BYTE   FART+FD,FRET    ;FINAL RESULT
0068    BD   0000G      ATN1:   JSR     DOFP
                                ;
                                ; USE APROX THAT ATN=X*(A/P1+A1*X 2/(B1+X 2+A2/(B2+X 2+A3/(B3+X 2+A4/(B4+X 2))))
                                ;
                                ;WHERE
                                ;A1=-.106418421649309E 01
                                ;A2=-.138394946555657E 02
                                ;A3=-.121932123923561E 01
                                ;A4=-.795076107678883E-01
                                ;B1= .482884240574443E 01
                                ;B2= .654757378157612E 01
                                ;B3= .214380802190815E 01
                                ;B4= .126686915230477E 01
                                ;
006E    00G  C0G    00G         .BYTE   FDUP,FDUP+FM,FSAV       ;SAVE X 2,LEAVE X ON STACK
0071    00G                     .BYTE   FHIM                    ;A1
0072    8401 8837   3038        .WORD   H8401, H8837, H3038, H6DEC
0078    6DEC
007A    00G                     .BYTE   FHIM                    ;A2
007B    8404 DD6E   91F2        .WORD   H8404, H0DD6E, H91F2, H0AD24
0081    AD24
0083    G0G                     .BYTE   FHIM                    ;A3
0084    8401 9C12   B7E6        .WORD   H8401, H9C12, H0B7E6, H0EAE5
008A    EAE5
008C    00G                     .BYTE   FHIM                    ;A4
008D    83FD A2D4   E2E0        .WORD   H83FD, H0A2D4, H0E2E0, H3FDA
0093    3FDA
0095    C0G                     .BYTE   FHIM                    ;B4
0096    0401 A228   C484        .WORD   H0401, H0A228, H0C484, H0BAE8
009C    BAE8
009E    80G  E0G                .BYTE   FRES+FA,FART+FD         ;A4/(B4+X 2)
00A0    80G                     .BYTE   FRES+FA
00A1    80G                     .BYTE   FHIM+FA                 ;ADD B3
00A2    0402 8934   268F        .WORD   H0402, H8934, H268F, H0BFDE
00A8    BFDE
00AA    E0G                     .BYTE   FART+FD                 ;DIVIDE INTO A3
00AB    80G                     .BYTE   FRES+FA                 ;ADD X 2
00AC    80G                     .BYTE   FHIM+FA                 ;ADD B2
00AD    0403 D185   B973        .WORD   H0403, H0D185, H0B973, H8087
00B3    8087
00B5    E0G                     .BYTE   FART+FD                 ;DIVIDE INTO A2
00B6    80G                     .BYTE   FRES+FA
00B7    80G                     .BYTE   FHIM+FA                 ;ADD B1
00B8    0403 9A85   E082        .WORD   H0403, H9A85, H0E082, H4841
00BE    4841
```

```
                00C0    E0G                     .BYTE   FART+FD                ;DIVIDE INTO A1
                00C1    C0G                     .BYTE   FRES+FM                ;MULT BY X 2
                00C2    80G                     .BYTE   FHIM+FA                ;ADD 4/PI
                00C3    0401    A2F9    836E    .WORD    0401, H0A2F9, H836E, H4E44
        00C9    4E44
                00CB    C0G                     .BYTE   FART+FM                ;MULT BY X
                00CC    00G                     .BYTE   FRET                   ;AND THATS ALL
                                                ;
                                                ;NOW COMPENSATE FOR RANGE REDUCTIONS
                00CD    76      0000G           ROR R1
                00D0    24      0D              BCC     ATNQ                   ;DONT NEED 90 DEG MOD
                00D2    BD      0000G           JSR     DOFP
                00D5    80G                     .BYTE   FHIM+FA                ;ADD 2/3 OCTANT
                00D6    0400    AAAA    AAAA    TWTHDS: .WORD    H0400, H0AAAA, H0AAAA, H0AAAA
        00DC    AAAA
                00DE    00G                     .BYTE   FRET
                00DF    76      0000G           ATNQ:   ROR     R1             ;CHECK FOR 90 DEG CORECTION
                00E2    24      0C              BCC     2$
                00E4    BD      0000G           JSR     FPNEG                  ;X=90DEG-X
                00E7    CE      0000G           LDX     FPTWO,I
                00EA    BD      0000G           JSR     PSHFPN
                00ED    BD      0000G           JSR     FPADD
                00F0    96      01G             2$:     LDA A   R1+1,D         ;CHECK - FLAG
                00F2    2A      03              BPL     3$
                00F4    BD      0000G           JSR     FPNEG
                00F7    96      00G             3$:     LDA A   CTKN,D         ;CHECK IF ARCCOS
                00F9    81      00G             CMP A   ACSCOD,I
                00FB    26      0C              BNE     4$                     ;NO, OK
                00FD    BD      0000G           JSR     FPNEG                  ;YES, X=2-X
                0100    CE      0000G           LDX     FPTWO,I
                0103    BD      0000G           JSR     PSHFPN
                0106    BD      0000G           JSR     FPADD                  ;2-X
                0109    FE      0000G           4$:     LDX     CONV1          ;CHANGE TO RADS,DEG,GRADS
                010C    BD      0000G           JSR     A8X                    ;INVERSE CONV 8 BYTES AFTER OTHERS
                010F    BD      0000G           JSR     PSHFPN
                0112    BD      0000G           JSR     FPMUL
                0115    7F      0000G           CLR     ERRCD
                0118    7E      0000G           JMP     RTRN                   ;RETURN
```

```
RANDOM NUMBER GENERATOR

                                        .SBTTL  RANDOM NUMBER GENERATOR
                                        .GLOBL  PIAKB
                                        .GLOBL  INTMLC,INTMLA,SHMR
                                        ;IF ARG >0, GET NEXT RANDOM NO
                                        ;IF ARG=0, RESET KERNEL
                                        ;IF 0>ARG)-1,SET KERNEL TO ARG
                                        ;IF ARG(-1, GET NEXT RANDOM

0118                                    NAME    RND
0118    BD    0000G                     JSR     PSHRET
011B    A6    03                        LDA A   3,X             ;TEST FOR ARG=0
011E    27    40                        BEQ     RND1
0120    A6    01                        LDA A   1,X             ;TEST SIGN BIT
0122    2A    40                        BPL     RND2            ;X>0
0124    85    04                        BIT A   4,I             ;TEST SIGN OF EXPONENT
0126    27    19                        BEQ     RNDS    ;0>X)-.5
0128    85    03                        BIT A   3,I
012A    26    15                        BNE     RNDS
012C    E6    02                        LDA B   2,X             ;REST OF EXPONENT
012E    27    11                        BEQ     RNDS            ;/X/<1

                                        ;/X/)=1,RANDOMIZE THE KERNEL
0130    FE    0005G                     LDX     KERNEL+5
0133    86    3F                        LDA A   77,I
0135    08                      1$:     INX
0136    B5    0000G                     BIT A   PIAKB
0139    26    FA                        BNE     1$
013B    FF    0005G                     STX     KERNEL+5
013E    30                              TSX
013F    20    23                        BRA     RND2
0141    A6    02              RNDS:     LDA A   2,X
0143    40                              NEG A
0144    81    30                        CMP A   48.,I           ;CHECK FOR EXP VERY SMALL
0146    24    18                        BCC     RND1            ;USE DEFAULT REALLY
0148    BD    0000G                     JSR     SHMR            ;NORMALIZE TO TRUE FRACTION
014B    CE    0000G                     LDX     KERNEL,I
014E    BD    0000G                     JSR     PULFPN          ;AND SET KERNEL
0151    76    0007G                     ROR     KERNEL+7        ;FORCE ODD
0154    0D                              SEC
0155    79    0007G                     ROL     KERNEL+7
                0400                    KEX= H00400     ;A 0 EXPONENT
0158    CE    0400                      LDX     KEX,I
015B    FF    0000G                     STX     KERNEL
015E    20    08                        BRA     RND3
0160    BD    0000G             RND1:   JSR     SETRND          ;SET DEFAULT
0163    30                              TSX
0164    BD    0000G             RND2:   JSR     A9X             ;PRUNE ARGUMENT
0167    35                              TXS
]]]]]
0168    CE    0000G             RND3:   LDX     R4,I            ;SAVE REGISTERS FOR EVALUATOR
016B    BD    0000G                     JSR     PSHFPN
016E    86    03                        LDA A   3,I

                                        ;NOW TAKE KERNEL=KERNEL*5 17 MOD 2 48
                                        ;NORMALIZE THIS CRUD TO GET THE RETURNED
                                        ;VARIABLE
                                        ;
                                        ;THE MULT IS DONE IN THREE PASSES
```

```
                                                        ;SUCH THAT NONE OF THE MULTIPLIERS
                                                        ;IS GREATER THAN 16 BITS
0172  97  00G             STA A  T1,D          ;SET COUNTER
0174  CE  3D09            LDX    5*5*5*5*5*5,I ;FOR 1ST 2 PASSES
0177  DF  00G       2$:   STX    T2,D
0179  DE  00G       1$:   LDX    T2,D          ;DO 16 BIT BY 48 BIT INTEGER MULT
017B  DF  00G             STX    R7,D
017D  FE  0006G           LDX    KERNEL+6
0180  BD  0000G           JSR    INTMLC
0183  DE  00G             LDX    R7,D
0185  FF  0006G           STX    KERNEL+6
0188  DE  00G             LDX    T2,D
018A  DF  00G             STX    R7,D
018C  FE  0004G           LDX    KERNEL+4
018F  BD  0000G           JSR    INTMLA
0192  DE  00G             LDX    R7,D
0194  FF  0004G           STX    KERNEL+4
0197  DE  00G             LDX    T2,D
0199  DF  00G             STX    R7,D
019B  FE  0002G           LDX    KERNEL+2
019E  BD  0000G           JSR    INTMLA
01A1  DE  00G             LDX    R7,D
01A3  FF  0002G           STX    KERNEL+2      ;WHEW, AT LAST
01A6  96  00G             LDA A  T1,D          ;COUNTER
01A8  4A                  DEC A
01A9  27  0A              BEQ    3$            ;DONE
01AB  97  00G             STA A  T1,D
01AD  4A                  DEC A                ;NOT DONE, CHECK IF PASS 2 OR 3
01AE  26  C9              BNE    1$            ;SAME FOR PASS 2
01B0  CE  0C35            LDX    5*5*5*5*5,I
01B3  20  C2              BRA    2$            ;CHANGE FOR PASS 3
01B5  CE  0000G     3$:   LDX    R4,I
01B8  BD  0000G           JSR    PULFPN        ;RESTORE REGISTERS
01BB  CE  0000G           LDX    KERNEL,I
01BE  BD  0000G           JSR    PSHFPN        ;STACK RETURN VARIABLE
01C1  7E  0000G           JMP    NORM          ;AND NORMALIZE


                          .GLOBL KERNEL,SETRND
```

SET FUZZ PARAMS

```
                                        .SBTTL  SET FUZZ PARAMS
                                        .GLOBL  TYPARG,ERDOMN,FUZA,FUZB,FIX1,ERFUZZ,SETARG
                                ;
                                ;SET FUZA TO NUMBER OF BITS FOR COMPARE,APROX LOG10(BASE2)
                                ;TIMES 1ST PARAM. SET FUZB TO SECOND PARAM.
                                ;
01C4                            NAME    SETFUZ
01C4  8D  0000G                 JSR     PSHRET
01C7  8D  0000G                 JSR     SETARG
01CA  8D  0000G                 JSR     TYPARG
01CD  27  15                    BEQ     1$              ;THERE WERE TWO ARGUMENTS
01CF  81  30                    CMP A   #30,I
01D1  27  08                    BEQ     2$              ;ONLY ONE
01D3  86  00G           19$:    LDA A   ERDOMN,I        ;DOMAIN ERROR
01D5  97  00G           20$:    STA A   ERRCD,D
01D7  7E  0000G                 JMP     RTRN
01DA  86  00G           18$:    LDA A   ERFUZZ,I
01DC  20  F7                    BRA     20$
01DE  CE  0000G         2$:     LDX     FPZERO,I        ;FLOATING POINT 0
01E1  8D  0000G                 JSR     PSHFPN          ;MAKE TWO ARGUMENTS
01E4  30                1$:     TSX
01E5  A6  01                    LDA A   1,X             ;TEST 2ND ARG >0
01E7  2B  F1                    BMI     18$
01E9  CE  0000G                 LDX     FUZB,I          ;ARG OK
01EC  8D  0000G                 JSR     PULFPN
01EF  30                        TSX
01F0  A6  01                    LDA A   1,X             ;1ST ARGUMENT
01F2  2B  E6                    BMI     18$
01F4  CE  021A'                 LDX     FUZCNV,I                ;LOG10/LOG2
01F7  8D  0000G                 JSR     PSHFPN
01FA  8D  0000G                 JSR     FPMUL
01FD  30                        TSX
01FE  8D  0000G                 JSR     FIX1            ;GET INTEGER PART
0201  A6  03                    LDA A   3,X             ;SEE IF HUGE
0203  27  04                    BEQ     3$
0205  86  FF                    LDA A   #255.,I
0207  20  04                    BRA     4$              ;YES,TAKE CARE OF IT
0209  A6  04            3$:    LDA A   4,X
0208  27  CD                    BEQ     18$             ;CANT ASK FOR 0 BITS
020D  B7  0000G         4$:     STA A   FUZA            ;SET NUMBER OF BITS TO COMPARE
0210  8D  0000G                 JSR     A9X
0213  35                        TXS
0214  7F  0000G                 CLR     ERRCD           ;TAKE CARE OF ANY POSSIBLE ERFXOV
0217  7E  0000G                 JMP     RTRN
021A  0402  D49A  7840  FUZCNV: .WORD   #0402,#0D49A,#7848,#0EC11
0220  EC11
;;;;;
                                .END
0001'
```

```
ADBANK RT-11 MMAC  VM02-10   14-OCT-76 01:27:42 PAGE 17+
SYMBOL TABLE

ACSCOD= XXXXXX G      ADBANK=   0000RG      ATNE     0023R      ATNQ     00DFR      ATN1     0068R
A8X   = XXXXXX G      A9X   = XXXXXX G      BANK  =   0000      CMPFPN= XXXXXX G      CONV1 = XXXXXX G
CTKN  = XXXXXX G      DEGCON= XXXXXX G      DOFP  = XXXXXX G      ERARC = XXXXXX G      ERDOMN= XXXXXX G
ERFUZZ= XXXXXX G      ERRCD = XXXXXX G      ERTRNG= XXXXXX G      FA    =   0080       FART  = XXXXXX G
FD    =   00E0       FDUP  = XXXXXX G      FHEX  = XXXXXX G      FHIM  = XXXXXX G      FHIN  = XXXXXX G
FIX1  = XXXXXX G      FLEX  = XXXXXX G      FLIN  = XXXXXX G      FLOAT2= XXXXXX G      FM    =   00C0
FPA   = XXXXXX G      FPADD = XXXXXX G      FPB   = XXXXXX G      FPC   = XXXXXX G      FPCMP = XXXXXX G
FPDIV = XXXXXX G      FPMUL = XXXXXX G      FPNEG = XXXXXX G      FPONE = XXXXXX G      FPSUB = XXXXXX G
FPTWO = XXXXXX G      FPZERO= XXXXXX G      FRES  = XXXXXX G      FRET  = XXXXXX G      FS    =   00A0
FSAV  = XXXXXX G      FSWP  = XXXXXX G      FUZA  = XXXXXX G      FUZB  = XXXXXX G      FUZCNV   021AR
GRDCON= XXXXXX G      INTMLA= XXXXXX G      INTMLC= XXXXXX G      JMPAX = XXXXXX G      KERNEL= XXXXXX G
KEX   =   0400       LOG2  = XXXXXX G      MAXANS= XXXXXX G      NGTAB = XXXXXX G      NORM  = XXXXXX G
NORMR = XXXXXX G      OVLN2 = XXXXXX G      OVLTN = XXXXXX G      PIAKB = XXXXXX G      PICON = XXXXXX G
PSHFPN= XXXXXX G      PSHRET= XXXXXX G      PULFPN= XXXXXX G      QACS     0000RG      QASN     0000RG
QATN     0020RG       QRND     0118RG       QSETFU   01C4RG      RADCON= XXXXXX G      RNDS     0143R
RND1     0162R        RND2     0166R        RND3     016AR       RTRN  = 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      SETARG= XXXXXX G      SETRND= XXXXXX G      SHMR  = XXXXXX G
SIGNS = XXXXXX G      SLN   = XXXXXX G      SQR   = XXXXXX G      SQR3     005AR       TABAD = XXXXXX G
TABPNT= XXXXXX G      TAN15 = XXXXXX G      TRUTH = XXXXXX G      TWTHDS   00D6R       TYPARG= XXXXXX G
T1    = XXXXXX G      T2    = XXXXXX G      T3    = XXXXXX G      T4    = XXXXXX G      ZANS  = XXXXXX G

  ABS.    0000     00
          0222     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2922. WORDS
.SY: ADBANK/C:DK1: SELCLI,ADBANK
```

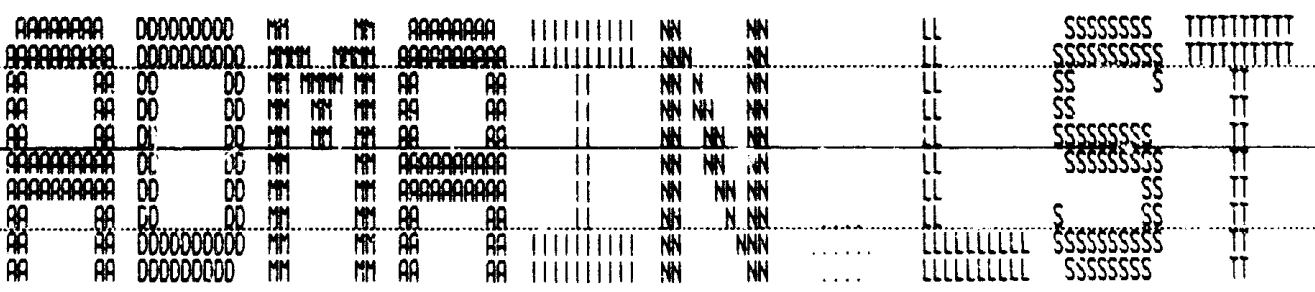
CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

KEX	16-41#	16-42												
LOG2	1-42#													
MAIN	1-57	1-69	2-1	3-1	4-1	5-9	5-10	5-33	6-1	7-1	8-1	9-1	10 1	11-1
	12-1	13-1	14-1	15-1	16-10	16-100	17-8							
MAXANS	1-31#													
NGTAB	1-37#													
NORM	1-40#	16-94												
NORMR	1-40#													
OVLN2	1-42#													
OVLTN	1-42#													
PIAKB	16-3#	16-26												
PICON	1-41#													
PSHFPN	1-31#	5-125	5-135	5-139	16-50	16-93	17-21	17-31						
PSHRET	1-31#	5-11	5-34	16-11	17-9									
PULFPN	1-31#	16-37	16-91	17-26										
QACS	5-9#	5-9#												
QASN	5-10#	5-10#												
QATN	5-33#	5-33#												
QRND	16-10#	16-10#												
QSETFU	17-8#	17-8#												
R0	1-35#													
R1	1-35#	5-35%	5-38%	5-50%	5-58%	5-115%	5-121%	5-127						
R2	1-35#													
R3	1-35#													
R4	1-35#	16-49	16-90											
R5	1-35#													
R6	1-35#													
R7	1-35#	16-65%	16-68	16-71%	16-74	16-77%	16-80							
RADCON	1-32#													
RND1	16-13	16-34	16-45#											
RND2	16-15	16-30	16-42#											
RND3	16-44	16-49#												
RNDS	16-17	16-19	16-21	16-31#										
RTRN	1-31#	5-142	17-17	17-45										
SETARG	17-3#	17-10												
SETRND	16-45	16-98#												
SHIR	16-4#	16-35												
SIGNS	1-40#													
SLN	1-37#													
SQR	5-3#	5-21												
SQR3	5-62#	5-66												
T1	1-35#	16-61%	16-82	16-85%										
T2	1-35#	16-63%	16-64	16-70	16-76									
T3	1-35#													
T4	1-35#													
TABAD	1-41#													
TABPNT	1-36#	5-55%												
TAN15	1-42#	5-54												
TRUTH	1-40#													
TWTHDS	5-119#													
TYPARG	17-3#	17-11												
ZANS	1-31#	5-26												

CROSS REFERENCE TABLE (CREF VM01-03)

NAME 1-20# 5-9 5-10 5-33 16-10 17-8
SEI 1-3#



.MAIN. RT-11 MMAC VM02-10 14-OCT-76 01:28:01
TABLE OF CONTENTS

```
                                .IDENT  /MEL126/

                        ; NOTE.  IF ETOX OR LOG IS IN THE XFN BANK THEN UP MUST!!!! BE ALSO.

                                .MACRO  NAME    N'
                                .IF     DF,MAIN
                                .GLOBL  NM
                        NM:
                                .ENDC
                                .IF     DF,BANK
                                .GLOBL  Q'NM
                        Q'NM:
                                .ENDC
                                .ENDM

                                .GLOBL  PSHRET,RTRN,PSHFPN,PULFPN,ERTRNG,ERRCD,ZANS,MAXANS
                                .GLOBL  CONV1,DEGCON,GRDCON,RADCON
                                .GLOBL  CTKN,JMPAX
                                .GLOBL  FPADD,FPSUB,FPMUL,FPDIV,A8X,A9X
                                .GLOBL  R0,R1,R2,R3,R4,F5,R6,R7,T1,T2,T3,T4
                                .GLOBL  FPA,FP8,FPC,CMPFPN,TABPNT
                                .GLOBL  FPNEG,SLN,NGTAB
                                .GLOBL  DOFP,FHEX,FHIN,FRET,FLEX,FLIN,FHIM,FART
                                .GLOBL  FSAV,FRES,FSWP,FDUP
                                .GLOBL  SIGNS,FLOAT2,NORM,NORMR,TRUTH,FPCMP
                                .GLOBL  FPZERO,FPONE,FPTWO,TABAD,PICON
                                GLOBL   OVLN2,LOG2,OVLTN,TAN15

                        ;BANK   =       0               ;   IF DEFINED GET BANK CODE
0000                    MAIN    =       0               ;IF DEFINED GET MAIN ROM CODE

                                .SBTTL  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                .SBTTL  XXXXXX                          XXXXXX

                                .IF     DF,BANK
                                .GLOBL  ADBANK
                        ADBANK  =       .
                                .SBTTL  XXXXXX  B A N K   V E R S I O N  XXXXXX
                                .TITLE  ADBANK
                                .ENDC

                                .IF     DF,MAIN
                                .GLOBL  ADMAIN
0000'                   ADMAIN  =       .
                                .SBTTL  XXXXXX  M A I N   V E R S I O N  XXXXXX
                                .TITLE  ADMAIN
                                .ENDC

                                .SBTTL  XXXXXX                          XXXXXX
                                .SBTTL  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

                                .NLIST  CND

]]]]]
                                .SBTTL  XXX DOFP        FLOATING POINT INTERPETER

                                ;FLOATING POINT INTERPETER
```

```
                                        ;MAKE LIFE EASIER FOR TRANSENDENTALS,GRAPHICS
                                        ;AND SO ON
                                        ;
                                        ;USED BY CALLING (JSR) DOFP,
                                        ;AND INSERTING CODES INLINE AFTER THE JSR
                                        ;
                                        ;EACH 8 BIT OP FEILD HAS A 5 BIT OPERATOR,
                                        ;EXECUTED BY TABLE, AND A 3 BIT FEILD,
                                        ;DEFINING ANY ARITH. OPERATION TO FOLLOW.

                                        .GLOBL  FAL,FS,FM,FD


                                        .EVEN
                    0000                FHEX=0
0000    DE      00G         FIPEX:  LDX     R0,D            ;PUSH EXTENDED
0002    EE      00          FIPEX1: LDX     0,X             ;GET ADRESS
0004    BD      0000G               JSR     PSHFPN
0007    DE      00G         FIPEX2: LDX     R0,D            ;BUMP PSUEDO PC BY 2
0009    08                          INX
000A    08                          INX
000B    DF      00G                 STX     R0,D
000D    20      35                  BRA     DOARTH          ;PROCESS OTHER 3 BITS OF OPCODE

                                        .EVEN
                    0008                FHIN=.-FIPEX/2      ;CODE FOR PUSH INDIRECT
0010    DE      00G                 LDX     R0,D
0012    EE      00                  LDX     0,X             ;DO INDIRECTION
0014    20      EC                  BRA     FIPEX1          ;LET ABOVE CODE DO WORK

                                        .EVEN
                    000B                FRET=.-FIPEX/2      ;CODE FOR RETURN TO ROUTINE
0016    DE      00G                 LDX     R0,D            ;HOW FAR WE ARE
0018    6E      00                  JMP     0,X             ; GO BACK THERE

                                        .EVEN
                    000D                FLEX=.-FIPEX/2      ;CODE FOR PULL EXTENDED
001A    DE      00G                 LDX     R0,D
001C    EE      00          FILEX1: LDX     0,X             ;GET ADRESS
001E    BD      0000G               JSR     PULFPN          ;DO PULL
0021    20      E4                  BRA     FIPEX2          ;INCRAMENT PSUEDO PC BY 2 AND CONTINUE

                                        .EVEN
                    0012                FSWP=.-FIPEX/2      ;CODE TO SWAP STACK ENTRIES
0024    20      56                  BRA     FSWP1
                                        .EVEN

                    0013                FLIN=.-FIPEX/2      ;CODE FOR PULL INDIRECT
0026    DE      00G                 LDX     R0,D            ;DO INDIRECTION
0028    EE      00                  LDX     0,X
002A    20      F0                  BRA     FILEX1

                                        .EVEN
                    0016                FDUP=.-FIPEX/2      ;CODE TO DUPLICATE ENTRY ON STACK
002C    30                          TSX
002D    08                          INX                     ;POINT X TO NUM. ON STACK
002E    20      05                  BRA     FRES1
```

```
                                .EVEN
                0018            FSAV=.-FIPEX/2          ;CODE OF SAVE IN TEMPORARY
0030    20      64              BRA     FSAV!
                                .EVEN
                0019            FRES=.-FIPEX/2          ;CODE FOR RESTORE FROM TEMP
0032    CE      0000G           LDX     FPA,I
0035    BD      0000G   FRES1:  JSR     PSHFPN
                001C            FART=.-FIPEX/2          ;JUST DO ARITHMATIC
0038    20      0R              BRA     DOARTH
                                .EVEN
                001D            FHIM=.-FIPEX/2          ;CODE FOR PUSH IMIDIATE
003A    DE      00G             LDX     R0,D
003C    BD      0000G           JSR     PSHFPN          ;PUSH
003E    BD      0000G           JSR     A8X             ;BUMP PC
0042    0F      00G             STX     R0,D            ;THIS CODE MUST BE EVEN # OF BYTES

0044    96      00G     DOARTH: LDA A   T4,D            ;RETRIVE CODE
0046    49                      ROL A
0047    24      21              BCC     NEXT            ;NOT SET, DO NOTHING
0049    49                      ROL A
004A    24      0C              BCC     ADSUB           ;ADD OR SUBTRACT CODE
004C    2B      05              BMI     1$
004E    BD      0000G           JSR     FPMUL
0051    20      17              BRA NEXT
0053    BD      0000G   1$:     JSR     FPDIV
0056    20      12              BRA     NEXT
0058    2B      05      ADSUB:  BMI     1$
005A    BD      0000G           JSR     FPADD
005D    20      08              BRA NEXT
005F    BD      0000G   1$:     JSR     FPSUB
0062    20      06              BRA     NEXT
0064    32              DOFP:   PUL A
0065    97      00G             STA A   R0,D
0067    32                      PUL A
0068    97      01G             STA A   R0+1,D
006A    DE      00G     NEXT:   LDX     R0,D            ;GET NEXT OP CODE, AND EXECUTE IT
006C    A6      00              LDA A   0,X
006E    08                      INX
006F    0F      00G             STX     R0,D            ;BUMP PSUEDO P.C.
0071    97      00G             STA A   T4,D            ;SAVE OP FOR LATER USE
0073    48                      ASL A
0074    84      3E              AND A   76,I            ;STRIP OFF UNUSED BITS
0076    CE      0000'           LDX     FIPEX,I
0079    7E      0000G           JMP     JMPAX           ;JUMP TO ROUTIENE VIA FUNNY JUMP
007C    CE      0000G   FSWP1:  LDX     FPC,I
007F    BD      0000G           JSR     PULFPN
0082    CE      0000G           LDX     FPB,I
0085    BD      0000G           JSR     PULFPN                  ;SAVE BOTH
0088    CE      0000G           LDX     FPC,I
008B    BD      0000G           JSR     PSHFPN                  ;PUT BACK IN REVERSE ORDER
008E    CE      0000G           LDX     FPB,I
0091    BD      0000G           JSR     PSHFPN
0094    20      AE              BRA     DOARTH                  ;AND GO ON
0096    CE      0000G   FSAV1:  LDX     FPA,I           ;TEMP IS FPA
```

XXX.DOFP FLOATING POINT INTERPETER

```
188  0099  80  0000G          JSR   PULFPN
189  009C  20  A6             BRA   DOARTH
190
192
193            0080           FA=200          ;ADD CODE
194            00A0           FS=240          ;SUB CODE
195            00C0           FM=300          ;MULT. CODE
196            00E0           FD=340          ;DIVIDE CODE
```

```
                                        .SBTTL  CONSTANTS FOR MATH ROUTINES

                                        ;STACK A PI
009E                                    NAME    PI
009E    CE  00A4'                       LDX     PICON+1
00A1    7E  0000G                       JMP     PSHFPN

00A4  0402  C90F  DAA2  PICON:  .WORD   H0402, H0C90F, H0DAA2, H2168
00AA  2168
00AC  0000  0000  0000  FPZERO: .WORD   0,0,0,0
00B2  0000
00B4  0402  8000  0000  FPTWO:  .WORD   H0402, H8000,0,0
00BA  0000
00BC  0401  B8AA  3B29  OVLN2:  .WORD   H0401, H0B8AA, H3B29, H5C17   ;1/LN 2
00C2  5C17
00C4  0400  B172  17F7  LOG2:   .WORD   H0400, H0B172, H17F7, H0D1CF   ;LN 2
00CA  D1CF
00CC  03FF  DE5?  D8A9  OVLTN:  .WORD   H03FF, H0DE58, HC08A9, H3728   ;1/LN 10
00D2  3728
00D4  00D6'                     TABAD:  .WORD   +2              ;ADRESS OF TABLE OF 2 1/16 M
00D6  0401  8000  0000  FPONE:  .WORD   H0401, H8000,0,0        ;2 0=1
00DC  0000
                                        ;
                                        ;THE 2 BYTES OF EXPONENT ARE MISSING FORM THE FOLLOWING
                                        ;TABLES. ALL EXPONENTS ARE 1. AND ARE CORRECTED
                                        ;FOR LATER IN THE EXP ROUTINE
00DE    85AA  C367  CC48                .WORD   H85AA, H0C367, H0CC48  ;2 1/16
00E4    8895  C1E3  EA88                .WORD   H8895, H0C1E3, H0EA88  ;2 2/16
00EA    91C3  D373  AB11                .WORD   H91C3, H0D373, H0AB11  ;2 3/16
00F0    9837  F051  8D88                .WORD   H9837, H0F051, H8D88   ;2 4/16
00F6    9EF5  3260  91A1                .WORD   H9EF5, H3260, H91A1    ;2 5/16
00FC    A5EE  D6A9  8151                .WORD   H0A5EE, H0D6A9, H08151 ;2 6/16
0102    AD58  3EEA  42A1                .WORD   H0AD58, H3EEA, H42A1   ;2 7/16
0108    8504  F333  F9DE                .WORD   H08504, H0F333, H0F9DE ;2 8/16
010E    BD08  A39F  580C                .WORD   H0BD08, H0A39F, H580C  ;2 9/16
0114    C567  2A11  5506                .WORD   H0C567, H2A11, H5506   ;2 10/16
011A    CE24  8C15  1F84                .WORD   H0CE24, H8C15, H1F84   ;2 11/16
0120    D744  FCCA  D69D                .WORD   H0D744, H0FCCA, H0D69D ;2 12/16
0126    E0CC  DEEC  2A94                .WORD   H0E0CC, H0DEEC, H2A94  ;2 13/16
012C    EAC0  C6E7  0D24                .WORD   H0EAC0, H0C6E7, H0D24  ;2 14/16
0132    F525  7D15  2486                .WORD   H0F525, H7D15, H2486   ;2 15/16
                                        ;TANGENT OF PI/12(15 DEGREES)
0138  03FF  8930  A2F4  TAN15:  .WORD   H03FF, H8930, H0A2F4, H0F66A
013E  F66A

                                        ;CONSTANTS USED BY SET RAD,GRAD,DEG
0140  03FB  B60B  60B6  DEGCON: .WORD   H03FB, H0B60B, H60B6, H0B61   ;8/360+1BIT FUDGE FACTOR
0146  0B61
0148  0406  B400  0000          .WORD   H0406, H0B400,0,0       ;360/8
014E  0000
0150  03FB  A3D7  0A3D  GRDCON: .WORD   H03FB, H0A3D7, H0A3D, H70A4   ;8/400+1 BIT CHEAT CONSTANT
0156  70A4
0158  0406  C800  0000          .WORD   H0406, H0C800,0,0               ;400/8
015E  0000
0160  0401  A2F9  836E  RADCON: .WORD   H0401, H0A2F9, H836E, H4E44   ;8/2PI
0166  4E44
0168  0400  C90F  DAA2          .WORD   H0400, H0C90F, H0DAA2, H2168           ;2PI/8
```

CONSTANTS FOR MATH ROUTINES

016E 2168
46

```
                                        .SBTTL  RELATIONAL OPERATORS
                                        .GLOBL  FUZZIE
                                        ;ALL RELATIONAL ROUTINES SET UP A CONDITION,
                                        ;AND CALL FUZZIE.  THEN THE OUTCOME IS COMPARED WITH
                                        ;THE CONDITION.
                                        ;1BIT=LT
                                        ;2BIT=EQ
                                        ;4BIT=GT
0170                                    NAME    NE
0170    86      05                      LDA A   5,I
0172    20      12                      BRA RELA
0174                                    NAME    EQ
0174    86      02                      LDA A   2,I
0176    20      0E                      BRA RELA
0178                                    NAME    GTE
0178    86      06                      LDA A   6,I
017A    20      0A                      BRA RELA
017C                                    NAME    LTE
017C    86      03                      LDA A   3,I
017E    20      06                      BRA RELA
0180                                    NAME    LT
0180    86      01                      LDA A   1,I
0182    20      02                      BRA RELA
0184                                    NAME    GT
0184    86      04                      LDA A   4,I
0186    97      00G     RELA:           STA A   R1,D            ;SAVE THE WHODUNIT
0188    BD      0000G                   JSR     PSHRET
018B    BD      0000G                   JSR     FUZZIE
018E    96      00G                     LDA A   R1,D            ;RETRIVE CONTINTION
0190    D6      00G                     LDA B   T4,D            ;WHAT WAS RELATION OF #'S
0192    26      04                      BNE     1$
                                ;WERE REALLY EQUAL
0194    85      02                      BIT A   2,I
0196    20      08                      BRA     4$
0198    2A      04              1$:     BPL     2$
019A    85      01                      BIT A   1,I
019C    20      02                      BRA     4$
019E    85      04              2$:     BIT A   4,I
01A0    27      0D              4$:     BEQ     NO              ;CONDITION DOSNT MATCH
01A2    30                              TSX                     ;YES IT DID, CHANGE 0 TO 1
01A3    86      04                      LDA A   4,I
01A5    A7      01                      STA A   1,X
01A7    86      01                      LDA A   1,I
01A9    A7      02                      STA A   2,X
01AB    86      80                      LDA A   200,I
01AD    A7      03                      STA A   3,X
01AF    7E      0000G   NO:             JMP     RTRN
```

```
                                        .SBTTL  TRIG SIN,COS,TAN ENTRY
                                        .GLOBL  COSCOD,TANCOD
                                        ;
                                ;DO RANGE REDUCTION FOR ALL TRIG FUNCTIONS
                                ;REDUCE INPUT TO OCTANTS, REMOVE EXCESS ROTATIONS
                                ;AND GO TO THE CORECT ROUTINE TO DO AN APROPRIATE REDUCTION

 01B2                                   NAME    TRIG
 01B2   BD  0000G                       JSR     PSHRET
 01B5   FE  0000G                       LDX     CONV1           ;POINTER TO (1/360,1/400,1/2PI)*8
 01B8   BD  0000G                       JSR     PSHFPN
 01BB   BD  0000G                       JSR     FPMUL           ;CONVERT TO OCTANTS
 01BE   30                              TSX
 01BF   A6  01                          LDA A   1,X             ;SIGN UPPER EXPONENT
 01C1   97  01G                         STA A   R1+1,D          ;SAVE SIGN
 01C3   84  7F                          AND A   177,I           ;FORCE POS
 01C5   A7  01                          STA A   1,X
 01C7   5F                              CLR B                   ;OCTANT
 01C8   81  04                          CMP A   4,I             ;TEST EXPONENT FOR + OR -
 01CA   27  09                          BEQ     OK1
 01CC   25  19                          BCS     OCO
 01CE   86  00G         SCTRNG: LDA A   ERTRNG,I
 01D0   97  00G                         STA A   ERRCD,D         ;SET ERROR FOR TRIG RANGE ERROR
 01D2   7E  0000G                       JMP     ZANS
 01D5   A6  02          OK1:    LDA A   2,X             ;REST OF EXPONENT
 01D7   81  14                          CMP A   20.,I           ;HOW MANY TIMES AROUND?
 01D9   24  F3                          BCC     SCTRNG          ;TOO MANY
 01DB   6F  02                          CLR     2,X             ;MAKE APROPRIATE EXPONENT
 01DD   BD  033C'                       JSR     SLN             ;SHIFT LEFT MANT ACCA TIMES
                                        ;BOTTOM 3 BITS OF B ARE OCTANT NUMBER
                                        ;SO THAT B IS NOW THE INTEGER PART OF THE FPN
 01E0   D7  00G                         STA B   T4,D            ;MUST RENORMALIZE FPN
 01E2   BD  0000G                       JSR     NORMP
 01E5   D6  00G                         LDA B   T4,D
 01E7   96  00G         OCO:    LDA A   CTKN,D          ;FIND OUT IF IT WAS A SIN,COS,OR TAN
 01E9   80  00G                         SUB  A  COSCOD,I
 01EB   25  06                          BCS     5$
 01ED   26  0D                          BNE     1$              ;WAS A SINE
 01EF   CB  02                          ADD B   2,I             ;COSINE,ROTATE 2 OCTANTS
 01F1   20  0F                          BRA     2$
 01F3   C5  02          5$:     BIT B   2,I             ;WAS A TAN
 01F5   27  0B                          BEQ     2$
 01F7   73  0001G                       COM     R1+1            ;OCT 2,3,6,OR 7, SO FLIP SIGN OF RESULT
 01FA   20  06                          BRA     2$
 01FC   96  01G         1$:     LDA A   R1+1,D          ;SINE, CHECK SIGN BIT
 01FE   2A  02                          BPL     2$              ;SIGN +, OK
 0200   CB  04                          ADD B   4,I             ;-, ROTATE 2 OCTS
 0202   56             2$:     ROR B                   ;ODD # OCTANT?
 0203   D7  00G                         STA B   R1,D
 0205   24  12                          BCC     3$              ;NO
 0207   C8  01                          EOR B   1,I             ;YES,S0 MOVE TO EQUIVALENT OCTANT
                                        ;THIS FLIPS SIN,COS BIT, AND
                                        ;THE TAN/COTAN BIT
 0209   D7  00G                         STA B   R1,D            ;LSB OF R1 IS 1 IF COTAN IS NEEDE
                                        ;THEESE ODD OCTNATS ALL GO BACKWARDS
                                        ;SO REALY NEED 1-X
 020B   CE  0006'                       LDX     FPONE,(         ;AND COMPLIMENT NO ON STACK
```

```
                020E    BD    0000G             JSR     PSHFPN
                0211    BD    0000G             JSR     FPSUB
                0214    30                      TSX
                0215    69    01                ROL     1,X             ;FLIP SIGN BIT +
                0217    64    01                LSR     1,X             ; SO RESULT IS X=1-X
                0219    96    00G       3$:     LDA A   CTKN,D
                021B    80    00G               SUB A   TANCOD,I
                021D    26    03                BNE     4$              ;CHECK IFF TANGENT
                021F    7E    02C3'             JMP     TAN             ;IT WAS
                0222    76    0000G     4$:     ROR     R1              ;SINE OR COSINE?
                0225    24    56                BCC     SPL
                0227    BD    0064'     CPL:    JSR     DOFP    ;COSINE APROXIMATION
                                                                        ;CHEBYSHEV APROX

                                        ;THE COSINE POLYNOMIAL IS A 12TH ORDER CHEBYSHEV OF THE FORM
                                        ;
                                        ;P(X)=(((((CG*X 2+CF)*X 2+CE)*X 2+CD)*X 2+CC)*X 2+CB)*X 2+CA
                                        ;
                                        ; WHERE
                                        ;
                                        ;CA= .999999999999996E0
                                        ;CB=-.499999999999991E0 *P'04 2
                                        ;CC= .416666666664705E-1*P104 4
                                        ;CD=-.138888888698159E-2*P104 6
                                        ;CE= .248015784673257E-4*P104 8
                                        ;CF=-.275552187227092E-6*P104 10
                                        ;CG= .206291063476645E-8*P104 12
                                        ;
                022A    D6                      .BYTE   FDUP+FM         ;FORM X 2
                0228    16                      .BYTE   FDUP
                022C    18                      .BYTE   FSAV
                022D    DD                      .BYTE   FHIM+FM         ;X 2*CG
                022E    03DF    F9E9    B910    .WORD   H03DF, H0F9E9, H0B910, H0F998
        0234    F998
                0236    9D                      .BYTE   FHIM+FA         ;ADD CF
                0237    83E7    D364    D940    .WORD   H83E7, H0D364, H0D940, H7395
        023D    7395
                023F    D9                      .BYTE   FRES+FM         ;MULT BY X 2
                0240    9D                      .BYTE   FHIM+FA         ;ADD CE
                0241    03EE    F0FA    7DA4    .WORD   H03EE, H0F0FA, H7DA4, H7742
        0247    7742
                0249    D9                      .BYTE   FRES+FM         ;MULT BY X 2
                024A    9D                      .BYTE   FHIM+FA         ;ADD CD
                024B    83F5    AAE9    E3ED    .WORD   H83F5, H0AAE9, H0E3ED, H0F5EF
        0251    F5EF
                0253    D9                      .BYTE   FRES+FM         ;MULT BY X 2
                0254    9D                      .BYTE   FHIM+FA         ;ADD CC
                0255    03F8    81E0    F84D    .WORD   H03F8, H081E0, H0F84D, H0D835
        0258    D835
                025D    D9                      .BYTE   FRES+FM         ;MULT BY X 2
                025E    9D                      .BYTE   FHIM+FA         ;ADD CB
                025F    83FF    9DE9    E64D    .WORD   H83FF, H09DE9, H0E64D, H0F228
        0265    F228
                0267    D9                      .BYTE   FRES+FM
                0268    9D                      .BYTE   FHIM+FA         ;FINALY ADD CA
                0269    0400    FFFF    FFFF    .WORD   H0400, H0FFFF, H0FFFF, H0FFFE
        026F    FFFE
```

```
                                        .BYTE   FRET            ;RETURN TO ASSEMBLER
0272    76  0000G      CJP:    ROR     R1              ;CHECK PLUS/MINUS BIT
0275    24  03         TJP:    BCC     SRET
0277    BD  0330'               JSR     FPNEG           ;NEGATE IF BIT WAS 1
027A    7E  0000G      SRET:   JMP     RTRN
027D    BD  0064'      SPL:    JSR     DOFP


                        ;THE SINE IS AN 11TH ORDER CHEBY APROXIMATION OF THE FORM
                        ;
                        ;P(X)=(((((SE*X^2+SE)*X^2+SD)*X^2+SC)*X^2+SB)*X^2+SA)*X
                        ;
                        ;WHERE
                        ;
                        ;SA= .999999999999972E0 *PI04
                        ;SB=-.166666666665404E0 *PI04^3
                        ;SC= .833333316960296-2*PI04^5
                        ;SD=-.198412607353790E-3*PI04^7
                        ;SE= .275548564509884E-5*PI04^9
                        ;SF=-.247320720952463E-7*PI04^11
                        ;
                        ; AND PI04=PI/4
                        ;
0280    16                      .BYTE   FDUP            ;LEAVE A COPY OF X AROUND
0281    06                      .BYTE   FDUP+FM         ;MAKE X^2
0282    16                      .BYTE   FDUP
0283    18                      .BYTE   FSAV            ;AND SAVE FOR LATTER USE
0284    DD                      .BYTE   FHIM+FM         ;MULT BY SF
0285    83E3    EE6D    D188            .WORD   H83E3, H0EE6D, H0D188, H7C19
0288    7C19
028D    9D                      .BYTE   FHIM+FA         ;ADD SE
028E    03E8    A838    40EE            .WORD   H03E8, H0A838, H40EE, H0C905
0294    AA05
0296    D9                      .BYTE   FRES+FM         ;MULT BY X^2
0297    9D                      .BYTE   FHIM+FA         ;ADD SD
0298    83F2    9969    61D5            .WORD   H83F2, H09969, H61D5, H0DD2C
029E    DD2C
02A0    D9                      .BYTE   FRES+FM         ;MULT BY X^2
02A1    9D                      .BYTE   FHIM+FA         ;ADD SC
02A2    03F8    A335    E336            .WORD   H03F8, H0A335, H0E336, H4C12
02A8    4C12
02AA    D9                      .BYTE   FRES+FM
02AB    9D                      .BYTE   FHIM+FA         ;ADD SB
02AC    83FD    A55D    E731            .WORD   ;83FD, H0A5ED, H0E731, H2891
02B2    2891
02B4    D9                      .BYTE   FRES+FM         ;MULT BY X^2
02B5    9D                      .BYTE   FHIM+FA         ;ADD SA AT LAST
02B6    0400    C90F    DAA2            .WORD   H0400, H0C90F, H0DAA2, H2162
02BC    2162
02BE    DC                      .BYTE   FART+FM         ;FINAL MULTIPLY BY X
02BF    08                      .BYTE   FRET            ;RETURN
02C0    7E  0272'               JMP     CJP             ;FINISH AND RETURN


                        ;TANGENT
                        ;USE APROCX
                        ;       (((T4*X^2+T3)*X^2+T2)*X^2+T1)*X
```

```
                                        ;TAN=  ----------------------------------
                                        ;       (((X 2+T8)XX 2+T7)XX 2+T6)XX 2+T5
                                        ;
                                        ;WHERE
                                        ;T1=  .10995878572142E08
                                        ;T2=-.90437476332431E06
                                        ;T3=  .14304200551812GE05
                                        ;T4=-.46836623610N669E02
                                        ;T5=  .14000387433524JE08
                                        ;T6=-.40301999917336 4E07
                                        ;T7=  .13659506248792BE06
                                        ;T8=-.10213175311147GE04
                                        ;
                                        ; INPUT X IN OCTANTS, 0<=X<=1
02C3    BD      0064'   TAN:    JSR     DOFP
02C6    16      D6              .BYTE   FDUP,FDUP+FM            ;GET X,X 2 ON STACK
02C8    16      18              .BYTE   FDUP,FSAV               ;SAVE X 2 IN TEMPORARY
02CA    DD                      .BYTE   FHIM+FM                 ;X 2*T4
02CB    8406    B758    B3DC    .WORD   H8406, H0B758, H0B3DC, H180C
02D1    180C
02D3    9D                      .BYTE   FHIM+FA                 ;ADD T3
02D4    040E    0F80    CD5D    .WORD   H040E, H00F80, H0CD5D, H7451
02DA    7451
02DC    D9                      .BYTE   FRES+FM                 ;XX 2
02DD    9D                      .BYTE   FHIM+FA                 ;ADD T2
02DE    8414    DCC8    6C36    .WORD   H8414, H0DCC8, H6C36, H92A1
02E4    92A1
02E6    D9                      .BYTE   FRES+FM                 ;XX 2
02E7    9D                      .BYTE   FHIM+FA                 ;ADD T1
02E8    0418    A7C8    A693    .WORD   H0418, H0A7C8, H0A693, H0BF9F
02EE    BF9F
02E0    DC                      .BYTE   FART+FM                 ;XX FOR NUMERATOR
02F1    19                      .BYTE   FRES                    ;START DENOMIN
02F2    9D                      .BYTE   FHIM+FA                 ;T8+X 2
02E3    840A    FF54    526E    .WORD   H840A, H0FF54, H526E, H0658
02F9    0658
02F8    D9                      .BYTE   FRES+FM                 ;(T8+X 2)XX 2
02FC    9D                      .BYTE   FHIM+FA                 ;ADD T7
02FD    0412    8564    C36C    .WORD   H0412, H08564, H0C36C, H58A1
0303    58A1
0305    D9                      .BYTE   FRES+FM                 ;XX 2
0306    9D                      .BYTE   FHIM+FA                 ;ADD T6
0307    8416    F5FB    DFF7    .WORD   H8416, H0F5FB, H0DFF7, H8905
0300    8905
030F    D9                      .BYTE   FRES+FM                 ;XX 2
0310    9D                      .BYTE   FHIM+FA                 ;FINISH BY ADDING T5
0311    0418    D5A1    036E    .WORD   H0418, H0D5A1, H036E, H0FB72
0317    FB72
0319    08                      .BYTE   FRET
031A    76      0000G           ROR     R1      ;CHECK ABOUT INVERT OF NUM,DENOM
031D    24      05              BCC     1$
031F    BD      00E4'           JSR     DOFP            ;SWAP
0322    12      0B              .BYT    FSWP,FRET
0324    BD      0000G   1$:     JSR     FPDIV           ;FORM QUOTIENT
0327    7F      0000G           CLR     EPRCD           ;IF TAN OF P1/2 DIV ZERO OCCURS
032A    79      0001G           ROL     R1+1            ;GET SIGN BIT
032D    7E      0275'           JMP     TJP             ;SET SIGN AND RETURN
```

ADMAIN RT-11 MACRO VM02-10 14-OCT-76 01:28:01 PAGE 4+
TRIG.SIN.COS.TAN ENTRY

209

```
                                        .SBTTL  NEG,SLN
0330    30          FPNEG:  TSX
0331    A6  05              LDA A   5,X         ;TEST IF 0
0333    27  06              BEQ     1$
0335    A6  03              LDA A   3,X         ; IF NO
0337    88  80              EOR A   200.!       ;FLIP SIGN BIT
0339    A7  03              STA A   3,X
033B    39          1$:     RTS                 ;BIG ROUTINE,HUH

                    ; SHIFT 48 BITS AT X+3 TO X+8 LEFT ACCA PLACES INTO ACCB

033C    5F          SLN:    CLR B
033D    40                  TST A
033E    27  10              BEQ     2$          ;IF 0, NO SHIFT
0340    68  08      1$:     ASL     8.,X        ;SHIFT ALL 48
0342    69  07              ROL     7,X
0344    69  06              ROL     6,X
0346    69  05              ROL     5,X
0348    69  04              ROL     4,X
034A    69  03              ROL     3,X
034C    59                  ROL B
034D    4A                  DEC A
034E    26  F0              BNE     1$
0350    39          2$:     RTS

                    ;TEST FOR TRUTH
0351    BD  0000G   TRUTH:  JSR     PSHRET
0354    A6  03              LDA A   3,X         ;TEST FOR 0,
0356    27  04              BEQ     1$          ;WHICH IS FALSE
0358    A6  01              LDA A   1,X
035A    46                  ROR A
035B    46                  ROR A
                            ;SIGN BIT OF EXP
                            ;IS 1 IF X<1/2
035C    BD  0000G   1$:     JSR     A9X
035F    35                  TXS                 ;PRUNE FPN
0360    84  01              AND A   1,!         ;A=1 IF /X/>=1/2
                            ;A=0 IF /X/<1/2
0362    7E  0000G           JMP     RTRN
```

```
                                        .SBTTL  SQUARE ROOT ROUTINE
                                        .GLOBL  SQR,ERSQR
                                        ;DO A SQUARE ROOT BY HALVING THE EXPONENT,
                                        ; AND DOING A 8 BIT TRUE SQUARE ROOT(WELL,REALLY 7 BIT)
                                        ; AND PUTTING IT IN THE UPPER BYTE OF THE MANTISA
                                        ; THEN 3 ITTERATIONS OF MR. NEWTON WILL GIVE FULL ACCURACY
                                        ; IN THE ANSWER.

0365  30              SQR:    TSX
0366  A6  05                LDA A   5,X             ;TEST FOR SQR OF 0
0368  26  01                BNE     1$
036A  39                    RTS                     ;RETURN, AS IS IDENTIT
036B  A6  03          1$:   LDA A   3,X
036D  2A  08                BPL     2$              ;TEST IF ARG >0
036F  C6  00G               LDA B   ERSQR,I         ;ERROR , <0
0371  D7  00G               STA B   ERRCD,D
0373  84  7F                AND A   127,I           ;FORCE PLUS
0375  A7  03                STA A   3,X
0377  BD  0000G       2$:   JSR     PSHRET
037A  BD  0064'             JSR     DOFP
037D  18  19  08            .BYTE   FSAV,FRES,FRET  ;SAVE COPY OF ARG
0380  30                    TSX
0381  E6  01                LDA B   1,X             ;1/2 EXPONENT. START BY SIGN EXTENSION
0383  CB  04                ADD B   4,I
0385  54                    LSR B
0386  66  02                ROR     2,X             ;ACTUAL DIVDE
0388  24  09                BCC     4$              ;EVEN, NO COMPENSATION
038A  64  03                LSR     3,X             ;ODD, 1/2 MANT
038C  66  04                ROR     4,X
038E  6C  02                INC     2,X             ;AND INC EXP
0390  26  01                BNE     4$
0392  5C                    INC B
0393  E7  01          4$:   STA B   1,X             ;MAKE LIKE A GOOD EXPONENT
0395  C6  07                LDA B   7,I
0397  D7  00G               STA B   T1,D            ;LOOP COUNTER
0399  5F                    CLR B                   ;RESULT REGISTER
039A  4F                    CLR A                   ;PARTIAL REMAINDER
                        ;DO LONG HAND DIVIDE IN BASE 2
039B  0D              SQLP: SEC
039C  59                    ROL B                   ;DOUBLE AND ADD 1 TO TRIAL "DIVISOR"
039D  68  04                ASL     4,X             ;GET 2 BITS FROM MANT INTO PARTIAL REMAINDER
039F  69  03                ROL     3,X
03A1  49                    ROL A
03A2  68  04                ASL     4,X
03A4  69  03                ROL     3,X
03A6  49                    ROL A
03A7  10                    SBA                     ;TRY IT
03A8  24  03                BCC     1$              ;OK
03AA  1B                    ABA                     ;NO GO, SO RESTORE
03AB  5A                    DEC B
03AC  5A                    DEC B                   ;TO MAKE UP FOR NEXT LINE
03AD  5C              1$:   INC B
03AE  7A  0000G             DEC     T1
03B1  26  E8                BNE     SQLP
03B3  5C                    INC B                   ;PSUEDO ROUND
03B4  E7  03                STA B   3,X             ;GUESS SQUARE ROOT
03B6  86  03                LDA A   3,I
```

```
    59  03B8  97  00G                  STA A   R1,D            ;COUNTER FOR NEWTON
    60  03BA  BD  0064'        SQLP2:  JSR     DOFP
    61  03BD  16  19   F2              .BYTE   FDUP,FRES,FSWP+FD       ;X/Y
    62  03C0  9C                       .BYTE   FART+FA         ;X/Y+Y
    63  03C1  08                       .BYTE   FRET
    64                      ;DEC EXPONENT TO GET (X/Y+Y)/2
    65  03C2  30                       TSX
    66  03C3  A6  02                   LDA A   2,X
    67  03C5  26  02                   BNE     1$
    68  03C7  6A  01                   DEC     1,X
    69  03C9  6A  02            1$:    DEC     2,X
    70  03CB  7A  0000G                DEC R1                  ;MORE TIMES?
    71  03CE  26  EA                   BNE     SQLP2           ;YES
    72  03D0  7E  0000G                JMP     RTRN
    73
```

```
                                        .SBTTL  EXP EXPONENTIAL ROUTINE
                                        .GLOBL  EREXP
                                        ;TAKE INPUT X, AND MULT BY 1/LOG 2
                                        ;THEN TAKE 2 X
                                        ;
                                        ;REDUCE RANGE BY X=+-(N+16(M+Y))
                                        ;WHERE N,M ARE INTEGERS, 0<=N<1024
                                        ;0<=M<16
                                        ;THEN
                                        ;
                                        ;2 X=2 N(EXPONENT ADD)*2 (M/16)(VIA TABLE LOOK UP)*(2 1/16) Y(VIA RATIONAL EXP)
                                        ;
03D3                                    NAME    ETOX
03D3  BD  0000G       ZZETOX:   JSR     PSHRET          ;IF THIS IS IN BANK UP MUST BE ALSO
03D6  CE  008C'                 LDX     OVLN2,I         ;1/LOG 2
03D9  BD  0000G                 JSR     PSHFPN
03DC  BD  0000G                 JSR     FPMUL
03DF  CE  0001                  LDX     1,I
03E2  DF    00G                 STX     R3,D            ;SET TEMP STORAGE
03E4  DF    00G                 STX     R1,D
03E6  30                        TSX
03E7  A6    01                  LDA A   1,X             ;SIGN TEST
03E9  97    01G                 STA A   R1+1,D          ;SAVE SIGN SO INVERT CAN BE DONE IF -
03EB  84    07                  AND A   7,I             ;ABS. VALUE
03ED  A7    01                  STA A   1,X
03EF  81    04                  CMP A   4,I             ;SIGN OF EXP
03F1  27    10                  BEQ     2$              ;LOOKS OK SO FAR
03F3  2B    38                  BMI     1$              ;EXP - SO NO INT PART
                                ;OTHERWISE EXP GROSSELY LARGE
03F5  86    00G         19$:    LDA A   EREXP,I         ;SORRY
03F7  97    00G                 STA A   ERRCD,D         ;SET EXP ARG TO LARGE
03F9  96    01G                 LDA A   R1+1,D          ;IS A 0 OR INF MOST APROPRIATE?
03FB  2A    03                  BPL     3$
03FD  7E  0000G                 JMP     ZANS
0400  7E  0000G           3$:   JMP     MAXANS
0403  A6    02           2$:    LDA A   2,X             ;REST OF EXPONENT
0405  81    09                  CMP A   9.,I            ;ANYTHING IN UPPER BYTE OF N?
0407  25    00                  BCS     5$              ;NO
0409  80    08                  SUB A   8.,I            ;HOW MANY BITS
040B  81    03                  CMP A   3,I             ;IF MORE THAN 2,
040D  24    E6                  BCC     19$             ;COMPLAIN
040F  BD  033C'                 JSR     SLN             ;GET THOSE BITS
0412  D7    00G                 STA B   R3,D
0414  86    08                  LDA A   8.,I
0416  BD  033C'           5$:   JSR     SLN             ;GET LOWER BYTE OF N
0419  5C                        INC B                   ;THE EXPONENTS OF ALL THE 2 M/16 ARE ONE, SO ADD
                                                        ;THE ONE HERE. WATCH FOR OVERFLOW
041A  26    09                  BNE     14$
041C  96    00G                 LDA A   R3,D
041E  4C                        INC A
041F  97    00G                 STA A   R3,D
0421  85    04                  BIT A   4,I             ;OVER INTO EXP SIGN BIT?
0423  26    D0                  BNE     19$             ;YUP
0425  D7    01G         14$:    STA B   R3+1,D
0427  86    04                  LDA A   4,I             ;SET UP NEXT 4 BITS FOR M
0429  20    11                  BRA     6$
042B  A7    01                  STA A   1,X
```

```
                                ;EXP NEG, ADD 4 TO IT, AND SEE IF IT GOES POS
                                ;FORCE NEW EXP TO 0
                                ;GET M
                                ;START APROX TO 2 Y/16
                                ;
                                ;
                                ;USE
                                ;2 (Y/16)=
                                ;
                                ;       (Q+R)/(Q-R)
                                ;
                                ;WHERE
                                ;Q=X 2+EX1
                                ;R=X*(X 2*EX2+EX3)
                                ;
                                ;EX1= .5728325426309896E+4
                                ;EX2= .3610070989487626E-2
                                ;EX3= .1154160545735176E+3
                                ;
                                ;ADD EX1
                                ;SAVE Q
                                ;X 2*EX2
                                ;ADD EX3
                                ;MULT BY X LEFT ON STACK
                                ;Q+R
                                ;R-Q, WHICH IS NEG
                                ;CHECK IF INVERSION IS NEEDED
                                ;EXCHANGE ARGS
                                ;FORM QUOTIENT
                                ;MAKE ADRESS INTO TABLE
                                ;MULT OFFSET BY 6
```

```
                                                CLR B
                                                ADD A   TABAD+1
                                                ADC B   TABAD
                                                STA A   TABPNT+1,D
                                                STA B   TABPNT,D
                                                LDX     TABPNT,D
                                                JSR     PSHFPN          ;PUSH WITH GARBAGE EXPONENT
                                                LDA A   R3,D
                                                LDA B   R3+1,D
                                                ;RESTORE A EXPONENT FROM R3, WHICH
                                                ;GIVES MULT BY 2 N BY ADD TO THE EXPONENT
                                                ;
                                                ;
                                                TSX
                                                STA B   2,X
                                                EOR A   4,I
                                                STA A   1,X
                                                LDA A   R1+1,D          ;TEST IF + OR -
                                                BPL     9$
                                                JSR     FPDIV
                                                BRA     10$
                                9$:             JSR     FPMUL
                                10$:            JMP     RTRN
```

```
                                        .SBTTL  LOGARITHMS
                                        .GLOBL  LGNCOD,ERLOG
                                        ;TAKE LOG OF X
                                        ;ERROR IF X<=0

                                        NAME    LGN
                                        NAME    LOG
04AD    BD    0000G     ZZLOG:  JSR     PSHRET          ;IF THIS IS IN BANK UP MUST BE
04B0    E6    03                LDA B   3,X             ;TEST FOR 0
04B2    27    04                BEQ     19$
04B4    A6    01                LDA A   1,X             ;TEST FOR NEG
04B6    2A    07                BPL     1$
04B8    86    00G       19$:    LDA A   ERLOG,I
04BA    97    00G               STA A   ERRCD,D         ;SET ERROR CODE
04BC    7E    0000G             JMP     RTRN
04BF                    1$:     ;
                        ;MAKE X=2 NOW WHERE
                        ;SQR(.5)<=W<SQR(2)
                        ;THEN   LOG(X)=LOG(W)+N*LOG(2)
                        ;SO STRIP OFF EXPONENT, AND SAVE AS A 2'S
                        ;COM NUMBER IN R3
04BF    80    04                SUB A   4,I             ;MAKE EXP 2'S COM
04C1    97    00G       2$:     STA A   R3,D
04C3    A6    02                LDA A   2,X
04C5    97    01G               STA A   R3+1,D
04C7    6F    02                CLR     2,X             ;SET EXP=0
04C9    86    04                LDA A   4,I
04CB    A7    01                STA A   1,X
                        ;THE NO. ON THE STACK IS NOW IN THE
                        ;RANGE OF .5 TO 1.  IF LESS THAN SQR(.5),
                        ;THEN DOUBLE IT AND DEC THE STORED EXPONENT
                        ;THE COMPARISON IS DONE ONLY WITH THE TOP 16
                        ;BITS OF THE MANTISSA.
                        ; THE RESULTING POSSIBLE ERROR IN THE RANGE WILL BE IGNORED
                        ; BY THE POLYNOMIAL WHICH PRODUCES THE LOG LATTER ON
04CD    A6    04                LDA A   4,X             ;MANT IN BA
04CF    80    05                SUB A    H05,I
04D1    C2    B5                SBC B    H0B5,I ;SQR(.5)= .B505
04D3    24    07                BCC     3$              ; > ALREADY, SO NO CHANGE
04D5    6C    02                INC     2,X             ;UPDATE EXPONENT
04D7    DE    00G               LDX     R3,D
04D9    09                      DEX
04DA    DF    00G               STX     R3,D            ;DEC TO COMPENSATE
04DC    BD    0064'     3$:     JSR     DOFP
                        ;
                        ;NOW LET T=(W-1)/(W+1)
04DF    16    A0                BYTE    FDUP,FHEX+FS    ;W-1
04E1    00D6'                   WORD    FPONE
04E3    12    80                BYTE    FSWP,FHEX+FA    ;W+1
04E5    00D6'                   WORD    FPONE
04E7    FC                      BYTE    FART+FD
                        ;NOW TAKE A CHEBY POLYNOMIAL IN T
                        ;LOG(W)=T*(((((((L7*T^2+L6)*T^2+L5)*T^2+L4)*T^2+L3)*T^2+L2)*T^2+
                        ;               L1)*T^2+2)
                        ;WHERE
                        ;
                        ;L1=.66666666666764
```

```
                                        ;L2=.399999999993023
                                        ;L3=.28571428726406432
                                        ;L4=.2222219705665668
                                        ;L5=.181836316888038
                                        ;L6=.153125279217173
                                        ;L7=.148097126899051
                                        ;
04E8    16      D6              .BYTE   FDUP,FDUP+FM
04EA    16      18              .BYTE   FDUP,FSAV       ;SAVE T 2
04EC    DD                      .BYTE   FHIM+FM         ;T 2%L7
04ED    03FE    9786    C6F2    .WORD    H03FE, H9786, H0C6F2, H0A6E5
04F3    A6E5
04F5    9D                      .BYTE   FHIM+FA         ;ADD L6
04F6    03FE    9CCC    DF89    .WORD    H03FE, H9CCC, H0DF89, H0B6E9
04FC    B6E9
04FE    D9                      .BYTE   FRES+FM         ;%T 2
04FF    9D                      .BYTE   FHIM+FA         ;ADD L5
0500    03FE    BA33    4CA9    .WORD    H03FE, H0BA33, H4CA9, H096F
0506    096F
0508    D9                      .BYTE   FRES+FM         ;%T 2
0509    9D                      .BYTE   FHIM+FA         ;ADD L4
050A    03FE    E38E    2800    .WORD    H03FE, H0E38E, H2800, H20FD
0510    20FD
0512    D9                      .BYTE   FRES+FM         ;%T 2
0513    9D                      .BYTE   FHIM+FA         ;ADD L3
0514    03FF    9249    24A2    .WORD    H03FF, H9249, H24A2, H8A04
051A    8A04
051C    D9                      .BYTE   FRES+FM
051D    9D                      .BYTE   FHIM+FA         ;ADD L2
051E    03FF    CCCC    CCCC    .WORD    H03FF, H0CCCC, H0CCCC, H0BD75
0524    BD75
0526    D9                      .BYTE   FRES+FM
0527    9D                      .BYTE   FHIM+FA ;ADD L1
0528    0400    AAAA    AAAA    .WORD    H0400, H0AAAA, H0AAAA, H0AAC6
052E    AAC6
0530    D9                      .BYTE   FRES+FM
0531    80                      .BYTE   FHEX+FA         ;ADD 2
0532    00B4'                   .WORD   FPTWO
0534    DC                      .BYTE   FART+FM         ;MULT BY T LEFT ON STACK
0535    08                      .BYTE   FRET
                                ;       NOW ADD IN LN2%R3
0536    D6      01G             LDA B   R3+1,D
0538    96      03G             LDA A   R3,D
053A    2A      05              BPL     LOGA
053C    43                      COM A                   ;TAKE ABS VALUE
053D    50                      NEG B
053E    26      01              BNE     LOGA
0540    4C                      INC A
0541    37              LOGA:   PSH B                   ;STACK INTEGER
0542    36                      PSH A
0543    36                      PSH A                   ;GARBAGE TAG
0544    BD      0000G           JSR     FLOAT2
0547    30                      TSX
0548    69      01              ROL     1,X
054A    79      0000G           ROL     R3
054D    66      01              ROR     1,X
```

```
 109  054F  CE  00C4'          LDX   LOG2,I
 110  0552  BD  0000G          JSR   PSHFPN
 111  0555  BD  0000G          JSR   FPMUL          ;MULT BY LN2
 112  0558  BD  0000G          JSR   FPADD
 113  055B  96    00G          LDA A CTKN,D         ;SEE IF LOG BASE 10
 114  055D  81    00G          CMP A LGNCOD,I
 115  055F  26    09           BNE   5$
 116  0561  CE  00CC'          LDX   ONLIN,I        ;MULT BY 1/LN10
 117  0564  BD  0000G          JSR   PSHFPN
 118  0567  BD  0000G          JSR   FPMUL
 119  056A  7E  0000G    5$:   JMP   RTRN
 120
```

```
                                        .SBTTL  SIGN,ABS VALUE FUNCTIONS
                                        .GLOBL  A11X
                        056D            NAME    SIG
                        056D    30      TSX                     ;SIGN FUNCTIONS RETRUNS A 0 FOR X=0,1 FOR X>1
                                                                ;AND -1 FOR X<0
                        056E    A6  05      LDA A   5,X         ;IS IT 0?
                        0570    26  01      BNE     1$
                        0572    39          RTS                 ;YES, IDENTITIY
                        0573    BD  0000G   1$:     JSR     PSHRET
                        0576    E6  01      LDA B   1,X         ;SIGN INTO TOP OF B
                        0578    BD  0000G   JSR     A9X         ;REMOVE AND REPLACE BY 1
                        057B    35          TXS
                        057C    CE  0006'   LDX     FPONE,I
                        057F    BD  0000G   JSR     PSHFPN      ;PUT ON A 1
                        0582    C4  80      AND B   200,I       ;SIGN IS STILL IN B
                        0584    CA  04      ORA B   4,I
                        0586    30          TSX
                        0587    E7  01      STA B   1,X
                        0589    7E  000G    JMP     RTRN

                        058C            NAME    ABS
                        058C    30      TSX
                        058D    A6  03      LDA A   3,X
                        058F    84  7F      AND A   177,I       ;REMOVE SIGN
                        0591    A7  03      STA A   3,X         ;
                        0593    39          RTS                 ;AND RETURN
```

```
                                        .SBTTL  LOGICAL OPERATORS-AND OR NOT
                                        .GLOBL  ORCOD
0594                                    NAME    NOT
0594    BD  0000G                       JSR     PSHRET
0597    BD  0351'                       JSR     TRUTH           ;FIND OUT IF TRUE
059A    4A                              DEC A                   ;AND  MAKE OPOSITE
059B    16                      LANS:   TAB
059C    CE  0006'                       LDX     #ONE,I
059F    BD  0000G                       JSR     PSHFPN          ;ASSUME TRUE, SET A ONE
05A2    5D                              TST B
05A3    26  03                          BNE     1$              ;AND IT WAS
05A5    7E  0000G                       JMP     ZANS
05A8    7E  0000G               1$:     JMP     RTRN
05AB                                    NAME    OR
05AB                                    NAME    AND
05AB    BD  C000G                       JSR     PSHRET
05AE    BD  0351'                       JSR     TRUTH
0581    97    00G                       STA A   T4,D
0583    BD  0351'                       JSR     TRUTH           ;TAKE OFF TWO OPERANDS
0586    D6    C0G                       LDA B   CTKN,D          ;FIND OUT IF AND OR OR
0588    C1    00G                       CMP B   ORCOD,I
058A    27    04                        BEQ     1$
058C    94    00G                       AND A   T4,D            ;AND , SO AND TRUTHS
058E    20    DB                        BRA     LANS
05C0    9A    00G               1$:     ORA A   T4,D            ;IT WAS A OR
05C2    20    D7                        BRA     LANS
```

```
                                        .SBTTL  FPCMP COMPARE NOS ON STACK

05C4  BD  0000G   FPCMP:  JSR     FPCHRET
05C7  A6  01              LDA A   1,X
05C9  2A  0E              BPL     1$              ;X)=0
                                                  ;
                                                  ;X=NO ON TOP OF STACK
                                                  ;Y=NEXT IN STACK
                                                  ;ROUTINE RETURNS -1 IF X)1 IN ACCA
                                                  ;1 IF Y)X
                                                  ;0 IF X=Y
05C8  E6  0A              LDA B   10.,X           ;KNOW X<0, TEST Y
05CD  2A  1A              BPL     YGT
05CF  BD  FFFFG           JSR     R9X-1           ;POINT AT Y, INC POINT BY 10
05D2  DF  00G             STX     TABPNT,D
05D4  BD  0000G           JSR     CMPFPN          ;COMPARE ABS VALUES
05D7  20  00              BRA     CMRET
05D9  A6  0A      1$:     LDA A   10.,X
05DB  2B  10              BMI     XGT             ;Y<0, Y)=0
05DD  BD  FFFFG           JSR     R9X-1           ;POINT TO Y
05E0  DF  00G             STX     TABPNT,D
05E2  BD  0000G           JSR     CMPFPN
05E5  40                  NEG A
05E6  7E  0000G   CMRET:  JMP     RTRN
05E9  86  01      YGT:    LDA A   1,I
05EB  20  F9              BRA     CMRET
05ED  86  FF      XGT:    LDA A   -1,I
05EF  20  F5              BRA     CMRET
05F1  4F          XYSAME: CLR A
05F2  20  F2              BRA     CMRET
```

```
                                        .SBTTL  UP  A B ROUTINE
                                        .GLOBL  ERUP,ERUPN
                                ;TEST IF EITHER A OR B IS 0, AND DO SOMETHING APROPRIATE IF IT IS
                                ;
05F4                                    NAME    UP
05F4  8D  0000G                         JSR     PSHRET
05F7  A6  0C                            LDA A   12.,X           ;CHECK A
05F9  26  0E                            BNE     UP10            ;FOR 0
05FB  A6  03                            LDA A   3,X             ;IT WAS, SO 0 0 IS NG
05FD  26  04                            BNE     UP01
05FF  86  000G                          LDA A   ERUP.I          ;IT WAS 0 0
0601  >.  000G                          STA A   ERRCD,D
0603  8D  0000G         UP01:          JSR     A9X
0606  35                               TXS                     ;C X=0
0607  20  75            UP11:          BRA     UPRT
0609  30                UP10:          TSX                     ;NOW TEST FOR B=0, AS A<>0
060A  E6  03                           LDA B   3,X
060C  26  0F                           BNE     UP12
060E  8D  0000G                        JSR     A9X
0611  8D  0000G                        JSR     A9X             ;X 0=1
0614  35                               TXS
0615  CE  0006'                        LDX     FPONE,I
0618  8D  000CG                        JSR     PSHEPN
061B  20  EA                           BRA     UP11            ;RETURN
061D  A6  01              UP12:        LDA A   1,X             ;NEITHER IS 0,TRY B AS AN INTEGER
                                ;IF B IS AN INTEGER, DO VIA MULTIPLIES
                                ; OTHER WISE ,USE LOG AND E X
                                ;
                                ;ACCEPT INTEGER LESS THAN 256 ONLY
061F  97  00G                           STA A   R3.D            ;FOR LATTER USE
0621  84  07                            AND A   7,I
0623  81  04                            CMP A   4,I
0625  26  5A                            BNE     NOINT           ;EITHER NEG OR TO BIG
0627  86  08                            LDA A   8.,I
0629  A0  02                            SUB A   2,X             ;TEST IF EXP IN RANGE 0-8
062B  25  54                            BCS     NOINT           ;NOWAY
062D  27  06                            BEQ     1$              ;8, SO INT PART ALREADY IN B
062F  54                  2$:           LSR B                   ;RENORMALIZE, AND MAKE SURE ITS AN INTEGER
0630  25  4F                            BCS     NOINT
0632  4A                                DEC A
0633  26  FA                            BNE     2$
0635  D7  00G             1$:           STA B   R1.D            ;SAVE INTEGER IF IT REALLY IS
0637  A6  04                            LDA A   4,X             ;CHECK IF REST OF FRACTION IS 0
0639  AA  05                            ORA A   5,X
063B  AA  06                            ORA A   6,X
063D  AA  07                            ORA A   7,X
063F  AA  08                            ORA A   8.,X
0641  26  3E                            BNE     NOINT           ;IT ISN'T
0643  97  01G                           STA A   R1+1,D          ;CLEAR A COUNTER OF BITS IN POWER
0645  A6  01              ISINT:        LDA A   1,X             ;SAVE SIGN OF B
0647  97  00G                           STA A   R3.D            ;FOR LATTER USE
0649  8D  0000G                        JSR     A9X             ;AND GET IT OFF THE STACK
064C  35                               TXS
                                ;
                                ;NOW FIND THE LEAST NUMBER OF MULTIPLIES
                                ;TO PRODUCE B INTEGER
                                ;USES MSA**, (MELANSON'S SNEAKY ALGORITHM NO. 15)
```

```
                                                        ;THIS WAS NOT MEANT TO CONFUSE
                                                        ;THE CASUAL READER. IT JUST HAPPENED
                                                        ;THAT WAY.
                                                        ;
                                                        ;EX: A 10
                                                        ;SQUARE A,DUP A 2 ON STACK, THEN SQUARE TWICE MORE
                                                        ;ONE MULTIPLY THEN GIVES A 10
                                                        ;
0640  74  0000G     1$:   LSR   R1                    ;GET NEXT BIT
0650  27  14              BEQ   2$                    ;READY FOR MULT.
0652  24  08              BCC   3$                    ;NO ENTRY THIS TIME
0654  7C  0001G           INC   R1+1                  ;NOTE ENTRY
0657  30                  TSX
0658  08                  INX
0659  BD  0000G           JSR   PSHFPN                ;SAVE ON STACK ONE MORE COPY
065C  30            3$:   TSX
065D  08                  INX
065E  BD  0000G           JSR   PSHFPN                ;SQUARE STACK ENTRY
0661  BD  0000G           JSR   FPMUL
0664  20  E7              BRA   1$
0666  96  01G       2$:   LDA A R1+1,D                ;ANY MULTIPLIES
0668  27  08              BEQ   4$                    ;NO
066A  BD  0000G       5$:   JSR   FPMUL               ;WELL, DO THEM
066D  7A  0001G           DEC   R1+1
0670  26  F8              BNE   5$
0672  96  00G       4$:   LDA A R3,D                  ;CHECK FOR B<0
0674  2A  08              BPL   UPRT
0676  BD  0064'           JSR   DOFP                  ;X=1/X
0679  00                  .BYTE FHEX
067A  0006'               .WORD FPONE
067C  F2  08              .BYTE FSWP+FD,FRET
067E  7E  0000G     UPRT: JMP   RTRN
0681  A6  0A        NOINI: LDA A 10.,X                ;CHECK IF A<0
0683  2A  06              BPL   2$
0685  86  00G             LDA A ERUPN,!               ;FATAL ERROR
0687  97  00G       1$:   STA A ERRCD,D
0689  20  F3              BRA   UPRT                  ;JMP RTRN
068B  BD  0064'     2$:   JSR   DOFP                  ;SWAP SO IN RIGHT ORDER
068E  12  08              .BYTE FSWP,FRET
0690  BD  04AD'           JSR   LOG                   ;LOG A
                          .GLOBL LOG
0693  BD  0000G           JSR   FPMUL
0696  BD  03D3'           JSR   ZZETOX                ;EXP(B*LOG(A))
0699  96  00G             LDA A ERRCD,D               ;CHECK ERRORS
0698  27  E1              BEQ   UPRT
0690  86  00G             LDA A ERUP,!                ;MAP ERRORS INTO MINE
069F  20  E6              BRA   1$                    ;STORE ERRCD AND JMP RTRN
```

```
                                        .SBTTL  MIN,MAX FUNCTIONS

                        06A1                    NAME    MIN
                        06A1    BD      0000G   JSR     PSHRET
                        06A4    BD      05C4'   JSR     FPCMP           ;FIND OUT WHICH MAX IS
                        06A7    2F      11      BLE     PRUNEX          ;X>=Y
                        06A9    30              XANS:   TSX
                        06AA    BD      FFFFG   JSR     A9X-1           ;POINT AT SECOND ONE
                        06AD    BD      0000G   JSR     PULFPN          ;AND COPY X INTO STACK
                        06B0    20      0D      BRA     MINEND          ;JMP RTRN
                        06B2                    NAME    MAX
                        06B2    BD      0000G   JSR     PSHRET
                        06B5    BD      05C4'   JSR     FPCMP
                        06B8    2B      EF      BMI     XANS            ;X<Y
                        06BA    30              PRUNEX: TSX
                        06BB    BD      0000G   JSR     A9X             ;PRUNE X AND RETURN WITH Y AS ANSWER
                        06BE    35              TXS
                        06BF    7E      0000G   MINEND: JMP     RTRN
```

```
                                        .SBTTL  INTEGER ROUTINE
                                        .GLOBL  INT
                                ;THIS ROUTINE IS MADE DIFFICULT DUE TO THE
                                ; NON-SYMETRIC DEFFINITION
                                ;INT(1.5)=1
                                ;INT(-1.5)=-2

06C2  BD  0000G   INT:  JSR     PSHRET
06C5  A6  01            LDA A   1,X             ;SIGN AND EXP
06C7  85  04            BIT A   4,I             ;SIGN OF EXPONENT
06C9  26  16            BNE     1$
06CB  40        3$:     TST A                   ;/X/<1
06CC  2B  03            BMI     2$
06CE  7E  0000G         JMP     ZANS
06D1  BD  0000G   2$:   JSR     A9X
06D4  35                TXS                     ;0>X)-1, ANSWER IS -1
06D5  CE  0006'         LDX     FPONE.I
06D8  BD  0000G         JSR     PSHFPN
06DB  BD  0330'         JSR     FPNEG
06DE  7E  0000G   19$:  JMP     RTRN
06E1  85  03      1$:   BIT A   3,I             ;IS EXP BIG?
06E3  26  F9            BNE     19$             ;YES, IDENTITY
06E5  E6  02            LDA B   2,X             ;REST OF EXPONENT
06E7  27  E2            BEQ     3$              ;EXP IS 0, HENCE /X/<1
06E9  50                NEG B
06EA  CB  30            ADD B   48.,I           ;48-EXP
06EC  24  F0            BCC     19$             ;EXPONENT WAS BIG
06EE  4F                CLR A                   ;ZERO OUT ACCB BITS
                                                ;OF THE MANTISA, NOTING
                                                ;IF ANYTHING REALLY HAPPENED
                                                ;IF ANYTHING GOT KILL, MUST
                                                ;SUB 1 IF ARG IS NEGATIVE
06EF  97  00G           STA A   R1,D
06F1  C0  08      4$:   SUB B   8.,I
06F3  2B  0C            BMI     10$             ;NO MORE BYTES OF 0
06F5  A1  08            CMP A   8.,X
06F7  27  05            BEQ     5$
06F9  A7  08            STA A   8.,X
06FB  7C  0000G         INC     R1              ;NOTE BYTE WAS NOT 0
06FE  09          5$:   DEX                     ;POINT AT NEXT BYTE
06FF  20  F0            BRA     4$
0701  CB  08      10$:  ADD B   8.,I            ;RESTORE
0703  27  10            BEQ     11$             ;NO ODD BITS AROUND
0705  43                COM A                   ;A=-1
0706  48        12$:  ASL A                     ;MAKE A MASK
0707  5A                DEC B
0708  26  FC            BNE     12$
070A  A4  08            AND A   8.,X            ;MASK IT OFF
070C  A1  08            CMP A   8.,X
070E  27  05            BEQ     11$
0710  7C  0000G         INC     R1
0713  A7  08            STA A   8.,X
0715  30         11$:   TSX
0716  A6  01            LDA A   1,X             ;NOTE IF IT WAS NEG
0718  2A  C4            BPL     19$
071A  96  00G           LDA A   R1,D
071C  27  C0            BEQ     19$             ;WAS AN EVEN INT
```

```
 59          071E    CE    0006'           LDX     FPONE,I
 60          0721    BD    0000G           JSR     PSHFPN
 61          0724    BD    0000G           JSR     FPSUB
 62          0727    20    B5              BRA     19$             ;JMP RTRN
 63
```

.INTEGER.ROUTINE

```
 97
 98                                             .GLOBL  KERNEL.SETRND
 99
101
102          0729    CE    0736'        SETRND: LDX     DEFRND,I
103          072C    BD    0000G                JSR     PSHFPN          ;SET DEFAULT
104          072F    CE    0000G                LDX     KERNEL,I
105          0732    BD    0000G                JSR     PULFPN
106          0735    39                         RTS
107          0736    0400  8765   ABCD  DEFRND: .WORD   H0400, H8765, H0ABCD, H1111   ;MUST BE ODD
     073C    1111
108
```

```
49
50                  0001'                       .END
```

```
ADMAIN RT-11 MMAC  VM02-10   14-OCT-76 01:28:01 PAGE 17+
SYMBOL TABLE

ABS      058CRG    ADMAIN=  0000RG    ADSUB    0058R     AND      05ABRG    APROX    0446R
A1X  = xxxxxx G    A8X   = xxxxxx G    A9X   = xxxxxx G    CJP      0272R     CMPFPN= xxxxxx G
CMRET    05E6R     CONV1 = xxxxxx G    COSCOD= xxxxxx G    CPL      0227R     CTKN  = xxxxxx G
DEFRND   0736R     DEGCON   0140RG    DOARTH   0044R     DOFP     0064RG    EQ       0174RG
EREXP = xxxxxx G   ERLOG = xxxxxx G    ERRCD = xxxxxx G    ERSQR = xxxxxx G   ERTRNG= xxxxxx G
ERUP  = xxxxxx G   ERUPN = xxxxxx G    ETOX     03D3RG    FA    =  0080 G    FART  =  001C G
FD    =  00E0 G    FDUP  =  0016 G    FHEX  =  0000 G    FHIM  =  001D G    FHIN  =  0008 G
FILEX1   001CR     FIPEX    0000R     FIPEX1   0002R     FIPEX2   0007R     FLEX  =  0000 G
FLIN  =  0013 G    FLOAT2= xxxxxx G    FM    =  00C0 G    FPA   = xxxxxx G   FPADD = xxxxxx G
FPB   = xxxxxx G   FPC   = xxxxxx G    FPCMP    05C4RG    FPDIV = xxxxxx G   FPMUL = xxxxxx G
FPNEG    0330RG    FPONE    0006RG    FPSUB = xxxxxx G    FPTWO    00B4RG    FPZERO   00ACRG
FRES  =  0019 G    FRES1    0035R     FRET  =  000B G    FS    =  00A0 G    FSAV  =  0018 G
FSAV1    0096R     FSWP  =  0012 G    FSWP1    007CR     FUZZIE= xxxxxx G   GRDCON   0150RG
GT       0184RG    GTE      0178RG    INT      06C2RG    ISINT    0645R     JMPAX = xxxxxx G
KERNEL= xxxxxx G   LANS     0598R     LGN      04ADRG    LGNCOD= xxxxxx G   LOG      04ADRG
LOGP     0541R     LOG2     00C4RG    LT       0180RG    LTE      017CRG    MAIN  =  0000
MAX      0682RG    MAXANS= xxxxxx G    MIN      06A1RG    MINEND   068FR     NE       0170RG
NEXT     006AR     NGTAB = xxxxxx G    NO       01AFR     NOINT    0681R     NORM  = xxxxxx G
NORMR = xxxxxx G   NOT      0594RG    OCO      01E7R     OK1      01D5R     OR       05ABRG
ORCOD = xxxxxx G   OVLN2    00BCRG    OVLTN    00CCRG    PI       009ERG    PICON    00A2G
PRUNEX   068AR     PSHFPN= xxxxxx G    PSHRET= xxxxxx G    PULFPN= xxxxxx G   RADCON   0160RG
RELA     0186R     RTRN  = xxxxxx G    RO    = xxxxxx G    R1    = xxxxxx G   R2    = xxxxxx G
R3    = xxxxxx G   R4    = xxxxxx G    R5    = xxxxxx G    R6    = xxxxxx G   R7    = xxxxxx G
SCTRNG   01CER     SETRND   0729RG    SIG      056DRG    SIGNS = xxxxxx G   SLN      033CRG
SPL      027DR     SQLP     0398R     SQLP2    038AR     SQR      0365RG    SRET     027AR
TABAD    00D4RG    TABPNT= xxxxxx G    TAN      02C3R     TANCOD= xxxxxx G   TAN15    0138RG
TJP      0275R     TRIG     0182RG    TRUTH    0351RG    T1    = xxxxxx G   T2    = xxxxxx G
T3    = xxxxxx G   T4    = xxxxxx G    UP       05F4RG    UPRT     067ER     UP01     0603R
UP10     0609R     UP11     0607R     UP12     061DR     XANS     06A9R     XGT      05EDR
XYSAME   05F1R     YGT      05E9R     ZANS  = xxxxxx G    ZZETOX   03D3R     ZZLOG    04ADR

  ABS    0000     00
         073E     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2579. WORDS
..SY:ADMAIN/C/OK1:SET/LL/GMAIN
```

CROSS REFERENCE TABLE (CREF V01-03)

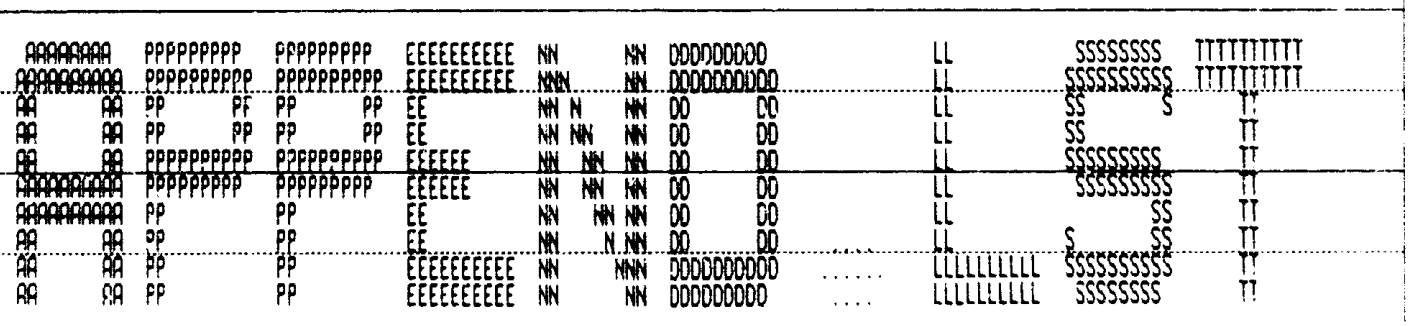
CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

PRUNEX	14-7	14-16#												
PSHFPN	1-31#	1-91	1-139	1-146	1-183	1-185	2-7	4-12	4-59	8-17	8-119	9-110	9-117	10-15
	11-10	13-24	13-73	13-76	15-19	15-60	16-103							
PSHRET	1-31#	3-28	4-10	6-28	7-20	8-15	9-9	10-10	11-5	11-17	12-4	13-7	14-5	14-13
	15-9													
PULFPN	1-31#	1-113	1-179	1-181	1-188	14-10	16-105							
R0	1-35#	1-89	1-92	1-95*	1-100	1-106	1-111	1-122	1-145	1-148*	1-166*	1-168*	1-169	1-172*
R1	1-35#	3-27*	3-30	4-16*	4-44*	4-46	4-50*	4-55*	4-68*	4-110*	4-201*	4-207*	7-59*	7-70*
	8-21*	8-24*	8-33	8-69*	8-103	8-108	8-110	8-130	13-43*	13-50*	13-67*	13-70*	13-79	13-82*
	15-34*	15-40*	15-52*	15-57										
R2	1-35#													
R3	1-35#	8-20*	8-44*	8-50	8-52*	8-55*	8-120	8-121	9-24*	9-26*	9-42	9-44*	9-94	9-95
	9-102*	13-31*	13-52*	13-84										
R4	1-35#													
R5	1-35#													
R6	1-35#													
R7	1-35#													
RADCON	1-32#	2-44#												
RELA	3-12	3-15	3-18	3-21	3-24	3-27#								
RTRN	1-31#	3-48	4-113	6-40	7-72	8-135	9-16	9-119	10-20	11-14	12-25	13-90	14-19	15-21
SCTRNG	4-23#	4-28												
SETRND	16-98#	16-102#												
SIG	10-4#	10-4#												
SIGNS	1-40#													
SLN	1-37#	4-30	6-13#	8-43	8-46	8-68								
SPL	1-69	4-114#												
SQLP	7-40#	7-55												
SQLP2	7-60#	7-71												
SQR	7-3#	7-10#												
SRET	4-111	4-113#												
T1	1-35#	7-36*	7-54*											
T2	1-35#													
T3	1-35#													
T4	1-35#	1-150	1-177*	3-31	4-33*	4-35	11-19*	11-24	11-26					
TABAD	1-41#	2-15#	8-114	8-115										
TABPNT	1-36#	8-116*	8-117*	8-118	12-16*	12-22*								
TAN	4-67	4-174#												
TAN15	1-42#	2-37#												
TANCOD	4-3#	4-65												
TJP	4-111#	4-208												
TRIG	4-9#	4-9#												
TRUTH	1-40#	6-28#	11-6	11-18	11-20									
UP	13-6#	13-6#												
UP01	13-11	13-14#												
UP10	13-9	13-17#												
UP11	13-16#	13-25												
UP12	13-19	13-26#												
UPRT	13-16	13-85	13-90#	13-95	13-103									
XANS	14-8#	14-15												
XGT	12-20	12-28#												
XYSAME	12-30#													
YGT	12-14	12-26#												
ZANS	1-31#	4-25	8-35	11-13	15-15									
ZZETOX	8-15#	13-101												
ZZLOG	9-9#													

CROSS REFERENCE TABLE (CREF V01-03)

NAME	1-20#	2-5	3-10	3-13	3-16	3-19	3-22	3-25	4-9	8-14	9-7	9-8	10-4	10-23
	11-4	11-15	11-16	13-6	14-4	14-12								
SEI	1-3#													



```
                                .IDENT  /MCD911/
                                .GLOBL  APPEND
0000'                           APPEND=.
                                .TITLE  APPEND
                                .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
                                .GLOBL  DELET,A13X,A6X,A7X,A9X,PSHRET,RTRN,PUSHXT,PULLRN
                                .GLOBL  SETARG,TYPARG,RTRNTG,LCLFLG,FIX1,ERRCD,ITM1TG,ITM2TG,ERAPP1
                                .GLOBL  VALTG,LNNOTG,GETSMO,APPOLD,RENUM,ERAPP2
                                .GLOBL  IMXTG,PGMPTR,PGMTG,NLPTR,A11X,A10X,HALTR,ERRCD8
                                .GLOBL  R12,R13,R14,R15,R16,R17,R18,R19,R20,ZX
                        ;
                        ;       APPEND PERFORMS THE APPEND STATEMENT -ISN'T THAT AMAZING
                        ;
                        ;       INPUT:  ARGUMENTS ON THE STACK
                        ;
                        ;       OUTPUT: (WITH A LOT OF LUCK AND COOPERATION FROM SCB)
                        ;               A PROGRAM FILE APPENDED INTO THE EXISTING PROGRAM
                        ;
                        ;       ERRORS: BAD ARGUMENTS
                        ;               ATTEMPTING TO APPEND AT A NONEXISTANT LINE NUMBER
                        ;               --SINCE APPEND CALLS THE I/O SYSTEM AND A NUMBER OF
                        ;                 OTHER THINGS A LOT OF ERRORS CAN BE RETURNED TO IT.
                        ;
                        ; REGISTER USAGE
                        ;       R0 - USED BY TYPARG
                        ;       R1 - USED BY TYPARG
                        ;       R2 - USED BY TYPARG
                        ;       R3 -
                        ;       R4 - RESULT OF TYPARG
                        ;       R5 -
                        ;       R6 -
                        ;       R7 -

0000    BD  0000G       JSR     PSHRET          ; SAVE THE RETURN ADDRESS
0003    4F              CLR A                   ;
0004    97      00G     STA A   R20,D           ;USED FOR FLAG
0006    DE      00G     LDX     ZX,D
0008    DF      00G     STX     NLPTR,D
000A    BD  0000G       JSR     SETARG          ; GET VALUE OF TOP ARG OF
                                                ; STACK
000D    26      23      BNE     OOPS1           ; IF NO VALUE, ERROR
000F    30              TSX
0010    BD  0000G       JSR     FIX1            ; INTEGERIZE ARG1
0013    26      1D      BNE     OOPS1
0015    DE      00G     LDX     R0,D            ; GET THE VALUE OF 2ND
0017    DF      00G     STX     R3,D            ; ARG ON STACK
0019    BD  0000G       JSR     TYPARG
001C    26      1E      BNE     DEFALT          ; IF NO 2ND ARG
001E    DE      00G     LDX     R3,D
0020    08              INX
0021    BD  0000G       JSR     FIX1            ; IF THERE WAS ONE INTEGERIZE IT
0024    26      0C      BNE     OOPS1
```

```
                                TSX
0026 30                         LDX     3,X
0027 EE 03                      STX     R13,D
0029 DF 00G                     TSX
002B 30                         JSR     A9X
002C 8D 0000G                   TXS
002F 35                         BRA     OK              ; IS THIS AN EOL
0030 20 12                      LDA A   ERRPP1,I        ; SET ERRCD AND QUIT
0032 86 00G             OOPS1:  STA A   ERRCD,D
0034 97 00G             EXIT:   JSR     H.LTR
0036 8D 0000G                   JMP     KTRN
0039 7E 0000G           DEFALT: LDX     10.,I
003C CE 000A                    STX     R17,D
003F DF 00G                     COM     R20
0041 73 0000F           OK:     TSX                     ; FIND THE LINE # AND
0044 30                         LDX     3,X             ; LOCATION OF THE FIRST LINE PREVIOUS
0045 EE 03                      STX     R12,D           ; TO THE POINT AT WHICH
0047 DF 00G                                             ; WE WILL BE APPENDING
                                PUL A
0049 32                         PUL A
004A 32                         JSR     GETSMA
004B 8D 0000G                   TSX                     ; ARE WE ATTEMPTING TO
004E 30                         LDX     1,X             ; APPEND AT A NON-EXISTANT
004F EE 01                      STX     R14,D
0051 DF 00G                     TSX
0053 30                         JSR     A10X
0054 8D 0000G                   TXS
0057 35                         LDX     R14,D
0058 DE 00G                     BEQ     WRONG
005A 27 06                      LDX     7,X
005C EE 07                      CPX     R12,D
005E 9C 00G                     BEQ     NEXT
0060 27 06              WRONG:  LDA A   ERRPP2,I
0062 86 00G                     STA A   ERRCD,D
0064 97 00G             CHAIN:  BRA     EXIT
0066 20 CE              NEXT:                           ; DELETE THE LINE AT WHICH WE ARE
0068                                                    ; APPENDING
                                STX     R3,D
0068 DF 00G                     STX     R4,D
006A DF 00G                                             ; FIND THE LINE # AND LOCATION OF
                                                        ; THE FIRST LINE FOLLOWING THE
                                                        ; POINT WE ARE APPENDING
                                LDX     R14,D
006C DE 00G                     LDX     3,X
006E EE 03                      LDA A   IMXTG,I
0070 86 00G                     JSR     PUSHXT
0072 8D 0000G                   LDX     R14,D
0075 DE 00G                     LDX     5,X
0077 EE 05                      BNE     1$
0079 26 03                      LDX     PGMPTR-3,I
007B CE FFFDG           1$:     LDA A   IMXTG,I
007E 86 00G                     JSR     PUSHXT
0080 8D 0000G                   JSR     DELET
0083 8D 0000G                   LDX     PGMPTR,D
0086 FE 00G                     LDA A   PGMTG,I
0088 86 00G                     JSR     PUSHXT
008A 8D 0000G                   LDX     ZX,D
008D DE 00G
```

```
                                STX     PGMPTR,D
                                JSR     APPOLD          ; INPUT THE PGM TO BE
                                JSR     PULLRN
                                .BYTE   36.
                                JSR     PULLRN
                                .BYTE   28.
                                JSR     PULLRN
                                .BYTE   32
                                LDX     R16,D
                                LDX     7,X
                                STX     R17,D
                                LDA A   ERRCD,D         ; APPENDED
                                BEQ     OK2
                                STA A   ERRCOB,D
                                CLR     ERRCD
OK2                             LDA A   #OFF,I ; PUSH A 2 16-1 AND CALL
                                PSH A                   ; GETSMA TO GET THE # OF
                                PSH A                   ; AND LOCATION OF THE LAST
                                LDA A   LNNOTG,I        ; LINE IN THE APPENDED PGM
                                PSH A
                                JSR     GETSMA
                                                        ;GET THE LINE # AND LOCATION OF THE
                                                        ;FIRST LINE IN THE APPENDED SSECTION
                                TSX
                                LDX     1,X
                                STX     R19,D
                                LDX     7,X
                                STX     R3,D
                                TSX
                                JSR     A6X
                                TXS
                                LDX     R13,D
                                STX     R1,D
                                LDX     PGMPTR,D
                                BEQ     NOREN
; IF THE INCREMENT WAS USER SUPPLIED, RENUMBER THE APPENDED
; SECTION REGARDLESS OF NEED.  IF NOT, DO THE RENUMBER ONLY
; IF THE FIRST LINE OF THE APPENDED SECTION DOES NOT MATCH THE LINE #
; AT WHICH WE ARE APPENDING
                                LDX     7,X
                                CPX     R12,D
                                BNE     RENNEW
                                LDA A   R20,D
                                BNE     NOREN
                                                        ; RENUMBER THE APPENDED SECTION
RENNEW:                         LDX     R12,D
                                STX     R0,D
                                                        ; STARTING WITH LINE #
                                                        ; AND USING INCREMENT
                                LDX     2X,D
                                STX     R2,D
                                JSR     RENUM
NOREN:                          LDX     PGMPTR,D
                                STX     R3,D
                                LDX     R18,D           ; RESTORE THE ORIGINAL
                                STX     PGMPTR,D        ; PROGRAM POINTER
                                LDX     R3,D
```

```
                00EE  DF  00G             STX   R18,D
                00F0  27  68              BEQ   EXIT1
                00F2  DE  00G             LDX   PGMPTR,D
                00F4  27  33              BEQ   ALMOST
                00F6  DE  00G             LDX   R19,D         ; GET THE (NEW) LINE NUMBER OF
                00F8  27  2F              BEQ   ALMOST
                00FA  EE  07              LDX   7,X           ; THE LAST LINE IN THE APPENDED SECTION
                00FC  DF  00G             STX   R0,D
                00FE  D6  01G             LDA B R0+1,D
                0100  96  00G             LDA A R20,D
                                                              ; IF THE INCR ON THE STACK
                0102  27  10              BEQ   RENOLD        ; WAS USER SUPPLIED GO DO THE
                0104  DE  00G             LDX   R16,D
                0106  27  21              BEQ   ALMOST
                0108  96  00G             LDA A R0,D          ; RENUMBER WHETHER OR NOT WE MUST
                010A  91  00G             CMP A R17,D
                010C  25  18              BCS   ALMOST        ; ELSE DO IT ONLY IF WE
                010E  22  04              BHI   RENOLD
                0110  D1  01G             CMP B R17+1,D       ; HAVE TO
                0112  25  15              BCS   ALMOST
                0114  96  00G   RENOLD:   LDA A R0,D
                0116  DB  01G             ADD B R13+1,D       ; RENUMBER THE ORIGINAL
                0118  99  00G             ADC A R13,D         ; PROGRAM SO THAT THE
                011A  D7  01G             STA B R0+1,D        ; STUFF AFTER THE APPEND
                011C  97  00G             STA A R0,D          ; POINT IS SLID DOWN BEYOND
                                                              ; THE HIGHEST LINE # IN
                                                              ; THE APPENDED SECTION
                011E  DE  00G             LDX   R17,D
                0120  DF  00G             STX   R2,D
                0122  DE  00G             LDX   R13,D
                0124  DF  00G             STX   R1,D
                0126  BD  0000G           JSR   RENUM
                0129              ALMOST:                     ; FORWARD PTR OF PREDECESSOR OF APPEND
                                                              ; POINT
                0129  96  00G             LDA A R18,D         ; = PTR TO FIRST LINE OF APPENDED
                012B  D6  01G             LDA B R18+1,D
                012D  DE  00G             LDX   R14,D
                012F  A7  03              STA A 3,X
                0131  E7  04              STA B 4,X
                0133  D6  01G             LDA B R14+1,D       ; BACK PTR OF FIRST LINE OF
                0135  96  00G             LDA A R14,D         ; APPENDED =  TO PREDECESSOR
                0137  26  01              BNE   1$
                0139  5F                  CLR B
]]]]]
                013A  DE  00G   1$:       LDX   R18,D         ; OF APPEND POINT
                013C  27  04              BEQ   2$
                013E  A7  05              STA A 5,X
                0140  E7  06              STA B 6,X
                0142              2$:                         ; BACK PTR OF SUCCESSOR OF APPEND
                0142  96  00G             LDA A R19,D         ; POINT = PTR TO HIGHEST LINE IN
                0144  D6  01G             LDA B R19+1,D       ; APPENDED SECTION
                0146  DE  00G             LDX   R16,D
                0148  27  04              BEQ   3$
                014A  A7  05              STA P 5,X
                014C  E7  06              STA B 6,X
                014E              3$:                         ; FORWARD PTR OF HIGHEST LINE IN
                014E  96  00G             LDA A R16,D         ; APPENDED SECTION= PTR TO SUCCESSOR
                0150  D6  01G             LDA B R16+1,D       ; OF APPEND POINT
```

```
 244  0152  DE   000G          LDX   R19,0
 245  0154  27   04            BEQ   EXIT1
 246  0156  A7   03            STA A 3,X
 247  0158  E7   04            STA B 4,X
 248  015A  96   000G    EXIT1:  LDA A  ERRCOB,0
 249  015C  7F   0000G         CLR   ERRCOB
 250  015F  97   000G          STA A ERRCO,0
 251                                        ; FIND RETURN ADDRESS IN
 252                                        ; STACK AND RETURN
 253  0161  7E   0036'          JMP   EXIT
 254        0001'               .END
```

```
APPEND RT-11 MACRO  VM02-10   14-OCT-76 01:28:33 PAGE 1+
SYMBOL TABLE

ALMOST   0129R       APPEND=   0000RG      APPOLD= 177777 G      A10X  = 177777 G      A11X  = 177777 G
A12X  = 177777 G     A6X   = 177777 G      A7X   = 177777 G      A9X   = 177777 G      CHAIN    0066R
DEFALT   003CR       DELET = 177777 G      ERAPP1= 177777 G      ERAPP2= 177777 G      ERRCD = 177777 G
ERRCOD= 177777 G     EXIT     0036R        EXIT1    015AR        F.X1  = 177777 G      GETSMA= 177777 G
HALTR = 177777 G     IMXTG = 177777 G      ITM1TG= 177777 G      ITM2TG= 177777 G      LCLFLG= 177777 G
LNNOTG= 177777 G     NEXT     0068R        NLPTR = 177777 G      NOREN    00E4R        OK       0044R
OK2      004FR       OOPSI    0032R        PGMPTR= 177777 G      PGMTG = 177777 G      PSHRET= 177777 G
PULLRN= 177777 G     PUSHXT= 177777 G      RENNEW   0009R        RENOLD   0114R        RENUM = 177777 G
RTRN  = 177777 G     RTRNTG= 177777 G      R0    = 177777 G      R1    = 177777 G      R10   = 177777 G
R11   = 177777 G     R12   = 177777 G      R13   = 177777 G      R14   = 177777 G      R15   = 177777 G
R16   = 177777 G     R17   = 177777 G      R18   = 177777 G      R19   = 177777 G      R2    = 177777 G
R20   = 177777 G     R3    = 177777 G      R4    = 177777 G      R5    = 177777 G      R6    = 177777 G
R7    = 177777 G     R8    = 177777 G      R9    = 177777 G      SETARG= 177777 G      TYPARG= 177777 G
VBLTG = 177777 G     WRONG    0062R        ZX    = 177777 G

  ABS.    0000      00
          0164      01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3208. WORDS
.SY:APPEND/C(DK1:SETCLI,APPEND
```

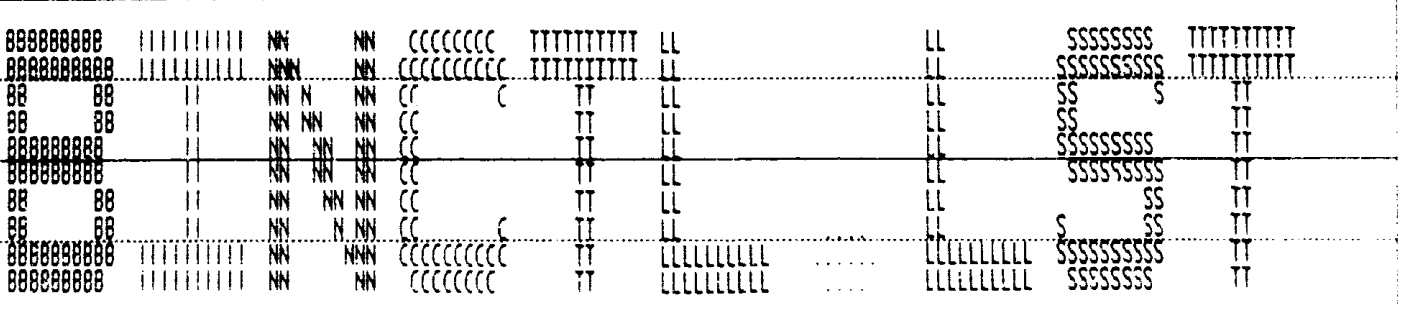
CROSS REFERENCE TABLE (CREF V01-03)

```
        1-18
A10X    1-24#   1-98
A11X    1-24#
A12X    1-21#
A6X     1-21#   1-159
A7X     1-21#
A9X     1-21#   1-77
ALMOST  1-190   1-192   1-200   1-203   1-206   1-219#
APPEND  1-17#   1-18#
APPOLD  1-23#   1-131
CHAIN   1-107#
DEFALT  1-68    1-84#
DELET   1-21#   1-126
ERAPP1  1-22#   1-80
ERAPP2  1-23#   1-135
ERRCD   1-22#   1-81*   1-106*  1-141   1-144*  1-250*
ERRCDB  1-24#   1-143*  1-248   1-249*
EXIT    1-82#   1-107   1-253
EXIT1   1-188   1-245   1-248#
FIX1    1-22#   1-63    1-71
GETSMA  1-23#   1-93    1-150
HALTR   1-24#   1-82
IMXTG   1-24#   1-117   1-123
ITM1TG  1-22#
ITM2TG  1-22#
LCLFLG  1-22#
LNNOTG  1-23#   1-148
NEXT    1-104   1-108#
NLPTR   1-24#   1-58*
NOREN   1-164   1-173   1-182#
OK      1-79    1-87#
OK2     1-142   1-145#
OOPSI   1-61    1-64    1-72    1-80#
PGMPTR  1-24#   1-122   1-126   1-130*  1-163   1-182   1-185*  1-189
PGMTG   1-24#   1-127
PSHRET  1-21#   1-54
PULLRN  1-21#   1-132   1-134   1-136
PUSHXT  1-21#   1-118   1-124   1-128
R0      1-20#   1-65    1-176*  1-194*  1-195   1-201   1-207   1-210*  1-211*
R1      1-20#   1-162*  1-217*
R10     1-20#
R11     1-20#
R12     1-25#   1-89*   1-103   1-170   1-175
R13     1-25#   1-75*   1-85*   1-161   1-208   1-209   1-216
R14     1-25#   1-96*   1-100   1-115   1-119   1-223   1-226   1-227
R15     1-25#
R16     1-25#   1-138   1-199   1-237   1-242   1-243
R17     1-25#   1-140*  1-202   1-205   1-214
R18     1-25#   1-184   1-187*  1-221   1-222   1-230
R19     1-25#   1-155*  1-191   1-235   1-236   1-244
R2      1-20#   1-180*  1-215*
R20     1-25#   1-56*   1-86*   1-172   1-196
R3      1-20#   1-66*   1-69    1-110*  1-157*  1-183*  1-186
R4      1-20#   1-111*
R5      1-20#
R6      1-20#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SEL 1-3#



14-OCT-76

TABLE OF CONTENTS

```
                                .TITLE  BINCTL--BINARY READ,WRITE CONTROL
                                .IDENT  /SBR017/
                        ;GOOD OLD GLOBALS
                        ;
                                .GLOBL  BINCTL
0000'                   BINCTL  =.
                                .GLOBL  A7X
                                .GLOBL  FILEIN,FILEOT,MTPOUT,MTPIN,IECIN,IECOUT
                                .GLOBL  XFRCTL
                                .GLOBL  A8X
                                .GLOBL  NIODEV
                                .GLOBL  DATIN
                                .GLOBL  GTDATA
                                .GLOBL  REASTG
                                .GLOBL  WRISTG
                                .GLOBL  REAVAL
                                .GLOBL  WRIVAL
                                .GLOBL  MTPTR
                                .GLOBL  SCRTCH
                                .GLOBL  CONCOD
                                .GLOBL  YAXIS
0001                            BINARY  =1
```

```
                                        .SBTTL  READHDR--READ HEADER
                                ;THIS ROUTINE READS IN THE HEADER FROM THE CURRENT I/O DEVICE
                                ;UPON EXIT:
                                ;       R0 CONTAINS HEADER STRIPPED OF HEADER BITS IF A STRING
                                ;       ACC A CONTAINS:
                                ;       0 IF VALUE
                                ;       1 IF STRING
                                ;       -2 IF ERROR
                                ;       -3 IF ERROR SET BY ROUTINE ACTIONS OTHER THAN READHDR
0000    DF      00G     BINBYT: STX     A.PTR,D         ; INTO R0
0002    DF      00G             STX     A.MAX,D
0004    BD      0062'           JSR     BININ
0007    96      00G             LDA A   ERRCD,D
0009    39                      RTS
                        ;
000A    CE      0000G   READHDR: LDX    R0,I            ; SET UP TO READ HEADER
000D    8D      F1              BSR     BINBYT          ; GET A BYTE
000F    9A      00G             ORA A   PNDEOF,D
0011    26      29              BNE     REAEOF
0013    D6      00G             LDA B   R0,D
0015    C1      FF              CMP B   255.,I          ; EOF?
0017    27      10              BEQ     2$
0019    CE      00G1G           LDX     R0+1,I
001C    8D      E2              BSR     BINBYT          ; GET ANOTHER BYTE
001E    26      1C              BNE     REAEOF
0020    4F              1$:     CLR A
0021    DE      00G             LDX     R0,D
0023    8C      2008            CPX     VULHDR,I        ; IS IT VALID HEADER FOR VALUE?
0026    26      17              BNE     RESTRC          ; TRY FOR A STRING
0028    39                      RTS
0029    86      01      2$:     LDA A   1,I
002B    D6      00G             LDA B   A.PRIM,D        ; IF MAGTAPE THEN SET EOF 0
002D    C1      23              CMP B   MTPD2,I
002F    26      19              BNE     RVER.
0031    DE      00G             LDX     MTPTR,D         ; SET MAGTAPE POINTER BACK A FEW
0033    09                      DEX
0034    DF      00G             STX     MTPTR,D ;
0036    D6      00G             LDA B   PNDEOF,D
0038    CA      80              ORA B   128.,I
003A    D7      00G             STA B   PNDEOF,D
003C    86      FD      REAEOF: LDA A   -3,I            ; SO WON'T SET ANY ERROR CODES
003E    39                      RTS
003F    4C              RESTRC: INC A
0040    D6      00G             LDA B   R0,D
0042    C8      40              EOR B   #40,I
0044    D7      00G             STA B   R0,D            ; DELETE HEADER INFO
0046    C5      FD              BIT B   #QEQ,I ; ALL BITS MUST=0 IF STRING
0048    27      09              BEQ     REXT            ; FOUND A STRING HEADER
004A    43              RVER.:  COM A                   ; SET ERROR
004B    81      FD      RVERR:  CMP A   -3,I            ; IF OTHER ROUTINES SET ERR
004D    27      04              BEQ     REXT            ; DON'T OVER RIDE IT
004F    C6      00G             LDA B   ERREAD,I
                                .GLOBL  ERREAD
0051    D7      00G             STA B   ERRCD,D
0053    39              REXT:   RTS
```

```
                                        .SBTTL  REAVAL--READ VALUE
0054    BD      B4      REAVAL: BSR     REAHDR          ; GET HEADER
0056    4D                      TST A                   ; MAKE SURE IT'S A VALUE
0057    26      F2              BNE     $VERR
0059    DE      00G             LDX     POINT.D         ; XFER VALUE TO MEMORY
005B    DF      00G             STX     A.PTR,D
005D    BD      0000G           JSR     A7X
0060    DF      00G             STX     A.MAX.D
                                                        ; USE NORMAL BUFFER HANDLERS

0062    CE      0069'   BININ:  LDX     BINITB,I        ; TABLE DISPATCH IT
0065    BD      0000G           JSR     XERCTL
0068    39                      RTS

0069    0000G   BINITB: .WORD   FILEIN          ; SPECIFIC DEVICE DRIVERS
0068    0000G           .WORD   IECIN
006D    0000G           .WORD   NIODEV
006F    0000G           .WORD   NIODEV
0071    0000G           .WORD   NIODEV
0073    0148'           .WORD   DATIN
0075    0000G           .WORD   MTPIN
0077    0000G           .WORD   NIODEV
```

```
                                        .SBTTL  REASTG--READ STRING
0029  8D  8F          REASTG: BSR     READHDR         ; GET HEADER
0078  81  01                  CMP A   1,I             ; A STRING?
007A  26  CC                  BNE     RVERR           ; SET ERROR
007F  DE  00G                 LDX     LENGTH,D
0081  96  00G                 LDA A   R0,D            ; LENGTH<--READ LEN.-REQ. LEN.
0083  D6  01G                 LDA B   R0+1,D
0085  D0  01G                 SUB B   LENGTH+1,D
0087  92  00G                 SBC A   LENGTH,D
0089  D7  01G                 STA B   LENGTH+1,D
008B  97  00G                 STA A   LENGTH,D
008D  2B  02                  BMI     1$              ; BRANCH IF REQ. LEN. > READ LEN.
008F  DF  00G                 STX     R0,D            ; SET POSIBLE READ LENGTH
0091  DE  00G           1$:   LDX     R0,D            ; UPDATE CHAR. COUNT
0093  DF  00G                 STX     CHRCNT,D
0095  27  4F                  BEQ     RSEXT           ; CHECK FOR ZERO LENGTH
0097  09                     DEX
0098  DF  00G                 STX     R0,D            ; MODIFY FOR I/O SYSTEM
009A  DE  00G                 LDX     POINT,D         ; MAKE POINTERS FOR READ
009C  DF  00G                 STX     A.PTR,D
009E  96  00G                 LDA A   POINT,D
00A0  D6  01G                 LDA B   POINT+1,D
00A2  DB  01G                 ADD B   R0+1,D
00A4  99  00G                 ADC A   R0,D
00A6  D7  01G                 STA B   A.MAX+1,D
00A8  97  00G                 STA A   A.MAX,D
00AA  8D  B6                  BSR     BININ           ; GO DO TRANSFER
00AC  96  00G                 LDA A   A.PRIM,D        ; IF DATA STATEMENT THEN DONE
00AE  81  22                  CMP A   DATDEV,I
00B0  27  34                  BEQ     RSEXT
00B2  DE  00G                 LDX     LENGTH,D
00B4  27  30                  BEQ     RSEXT
00B6  2B  2E                  BMI     RSEXT           ;SO IT WON'T TAKE 5 SECONDS TO RETURN
00B8  09                      DEX
00B9  DF  00G                 STX     LENGTH,D
00BB  96  00G                 LDA A   LENGTH,D        ; DONE?
00BD  27  13          RSLGS:  BEQ     RSSMS           ; READ SMALL STRING BUFFER
00BF  4A                      DEC A
00C0  97  00G                 STA A   LENGTH,D        ; READ FULL 256 BYTES
00C2  CE  0000G               LDX     SCRTCH,I        ; INTO SCRATCH (WOM)
00C5  DF  00G                 STX     A.PTR,D
00C7  CE  00FFG               LDX     SCRTCH+255.,I
00CA  DF  00G                 STX     A.MAX,D
00CC  8D  94                  BSR     BININ
00CE  96  00G                 LDA A   LENGTH,D
00D0  20  E8                  BRA     RSLGS
00D2  96  01G           RSSMS: LDA A  LENGTH+1,D      ; READ SMALL STRING INTO WOM
00D4  CE  0000G               LDX     SCRTCH,I
00D7  DF  00G                 STX     A.PTR,D
00D9  9B  01G                 ADD A   A.PTR+1,D       ; MOVE THE REMAINING BYTES INTO SCRATCH
00DB  97  01G                 STA A   A.MAX+1,D
00DD  96  00G                 LDA A   A.PTR,D
00DF  89  00                  ADC A   0,I
00E1  97  00G                 STA A   A.MAX,D
00E3  BD  0062                JSR     BININ
00E6  39              RSEXT:  RTS
```

```
                                        .SBTTL  WRISTG--WRITE STING
                                        .IF     NDF     BINARY
                                NBINFN:
                                WRISTG: JMP     NONCE           ; NO FUNCTION
                                        .GLOBL  NONCE
                                        .ENDC
                                        .IF     DF      BINARY
00E7    96      00G             WRISTG: LDA A   POINT,D         ; SET UP END LENGTH
00E9    D6      01G                     LDA B   POINT+1,D
00EB    D8      01G                     ADD B   LENGTH+1,D
00ED    99      00G                     ADC A   LENGTH,D
00EF    97      00G                     STA A   R0,D
00F1    D7      01G                     STA B   R0+1,D
00F3    96      00G                     LDA A   LENGTH,D        ; TEST IF STRING TO LONG TO BE WRITTEN
00F5    85      E0                      BIT A   340,I           ;UPPER 3 BITS MUST BE OFF
00F7    26      1B                      BNE     WRIERR          ; IF NOT SET ERROR
00F9    8B      40                      ADD A   SRTHDR,I        ; ADD IN STRING HEADER
00FB    97      00G                     STA A   LENGTH,D
00FD    CE      0000G                   LDX     LENGTH,I        ; GET ADDRESS OF LENGTH AND
0100    DF      00G                     STX     A.PTR,D         ; AND SET IT UP FOR TRANSFER
0102    08                              INX
0103    DF      00G                     STX     A.END,D
0105    8D      25                      BSR     BINOUT
0107    DE      00G                     LDX     R0,D            ; SET UP FOR TRANSFER OF REAL DATA
0109    9C      00G                     CPX     POINT,D         ; CHECK FOR ZERO LENGTH STRING
010B    27      08                      BEQ     WRIXXX
010D    09                              DEX                     ; MODIFY POINTER FOR I/O SYSTEM
010E    DF      00G                     STX     A.END,D
0110    20      20                      BRA     WRIPAT
                                        .BLKB   2               ; SPACE TAKER
0114    86      00G             WRIERR: LDA A   ERSTGL,I        ; STRING TOO LONG
                                        .GLOBL  ERSTGL
0116    97      00G                     STA A   ERRCD,D
0118    39                      WRIXXX: RTS
                                        .ENDC
```

```
                                        .SBTTL  WRIVAL--WRITE VALUE
                                        .IF     NDF,BINARY
                                WRIVAL: BRA     NBINFN
                                        .ENDC
                                        .IF     DF,BINARY
0119  CE  0190'                 WRIVAL: LDX     VLUHDR,I        ; GET ADDRESS OF HEADER INFO
011C  DF    00G                         STX     A.PTR,D         ; SET UP FOR XFER
011E  08                                INX
011F  DF    00G                         STX     A.END,D
0121  8D    09                          BSR     BINOUT
0123  DE    00G                         LDX     POINT,D         ; SET UP POINTERS FOR REMAINDER OF DATA
0125  DF    00G                         STX     A.PTR,D
0127  BD  0000G                         JSR     A7X
0129  DF    00G                         STX     A.END,D
012C  CE  0138'                 BINOUT: LDX     BINOTB,I
012F  7E  0000G                         JMP     XFRCTL

0132  DE    00G                WRIPAT: LDX     POINT,D         ; SET UP I/O SYSTEM POINTERS
0134  DF    00G                         STX     A.PTR,D
0136  20    E4                         BRA     BINOUT

0138  0000G                     BINOTB: .WORD   FILEOT          ; INDIVIDUAL DEVICE DRIVERS
013A  0000G                             .WORD   IECOUT
013C  0000G                             .WORD   NIODEV
013E  0000G                             .WORD   NIODEV
0140  0000G                             .WORD   NIODEV
0142  0000G                             .WORD   NIODEV
0144  0000G                             .WORD   MTPOUT
0146  0000G                             .WORD   NIODEV
                                        .ENDC
```

```
                                        .SBTTL  DATIN---DATA STATEMENT INPUT
                                        .GLOBL  ITGCOD,FLOAT1,PULF2N
0148  96  000G    DATIN:  LDA A   YAXIS,D         ; IF 0 THEN FIRST ACCESS MODE
014A  81  01              CMP A   1,I
014C  22  44              BHI     DATSEC
014E  4C                  INC A
014F  97  000G            STA A   YAXIS,D
0151  4A                  DEC A
0152  26  38              BNE     DATEXT
0154  BD  0000G           JSR     GTDATA          ; TRY TO FIND SOME DATA
0157  96  000G            LDA A   ERRCOD,D        ; ERRORS?
0159  26  34              BNE     DATEXT
015B  DE  000G            LDX     RO,D            ; GET POINTER TO CODE
015D  A6  00              LDA A   0,X             ; GET CODE TYPE
015F  81  000G            CMP A   ITGCOD,I        ; IS IT SPECIAL INTEGER?
0161  26  12              BNE     DATOTH
0163  A6  02              LDA A   2,X             ; IF SO PUT INTEGER ON STACK AND FLOAT IT
0165  36                  PSH A
0166  A6  01              LDA A   1,X
0168  36                  PSH A
0169  36                  PSH A
016A  BD  0000G           JSR     FLOAT1
016D  CE  0000G           LDX     R3,I            ; PUL IT INTO THE PSUEDO AREA
0170  BD  0000G           JSR     PULFPN
0173  20  05              BRA     DATVLC
0175  81  00G     DATOTH: CMP A   CONCOD,I
0177  26  09              BNE     DATSTR
0179  08                  INX
017A  DF  00G     DATVLC: STX     R1,D            ; SAVE IT AS POINTER TO VALUE
017C  CE  2008            LDX     VULHDR,I        ; SET UP FAKE HEADER
017F  DF  00G             STX     RO,D            ; CHEAT AND PUT IT WHERE IT GOES
0181  39                  RTS
0182  A6  02     DATSTR: LDA A   2,X             ; GE LENGTH (MUST BE <= 72)
0184  97  01G             STA A   RO+1,D
0186  86  40              LDA A   SRTHDR,I
0188  97  00G             STA A   RO,D
018A  08                  INX
018B  08                  INX
018C  08                  INX
018D  DF  00G             STX     R1,D            ; SET POINTER TO FIRST CHAR.
018F  39          DATEXT: RTS
                  VULHDR  =  H2008
                  SRTHDR  =  H40
0190  2008        VULHDR: .WORD   VULHDR
0192  7F  0000G   DATSEC: CLR     YAXIS           ; RESET MODE FLAG
0195  DE  00G     DATSE.: LDX     R1,D            ; GET CHAR. POINTER
0197  A6  00              LDA A   0,X             ; GET CHAR
0199  08                  INX
019A  DF  00G             STX     R1,D            ; SAVE NEW CHAR POINTER
019C  DE  00G             LDX     A.PTR,D         ; GET RESULT
019E  A7  00              STA A   0,X             ; STORE CHAR.
01A0  9C  00G             CPX     A.MAX,D         ; DONE?
01A2  27  05              BEQ     1$
01A4  08                  INX
01A5  DF  00G             STX     A.PTR,D
01A7  20  EC              BRA     DATSE.
01A9  39          1$:     RTS
```

```
                                        .SBTTL  TYPE----THE TYPE FUNCTION
                                ;THIS ROUTINE IS CALLED WITH AN ARGUMENT ON THE STACK AND RETURNS
                                ;WITH THE RESULT.
                                ;
                                ;THE TYPE FUNCTION RETURNS:
                                ;       0- FILE NOT OPEN OR INVALID HEADER
                                ;       1- EOF     END OF FILE
                                ;       2- ASCII FILE
                                ;       3- BINARY NUMBER
                                ;       4- BINARY STRING
                                ;
                                        .GLOBL  TYPE
                                        .GLOBL  PSHRET,RTRN
                                        .GLOBL  INITMT,CRTRST
                                        .GLOBL  A9X
                                        .GLOBL  MTRBFR          ; ROUTINE TO READ IN NEW MAGTAPE BUFFER
                                        .GLOBL  MTOPEN, MTSNEW
                                        .GLOBL  FIX1,MTSTT2,MTSRD
                                        .GLOBL  TYPFIL          ; FILE SYSTEM PROCESSOR
                                        .GLOBL  MTSTAT,MTSBIN,MTSASC,MTPTR,FLOAT1
                                ;
01AA    BD      0000G   TYPE:   JSR     PSHRET          ; SAVE RETURN ADDRESS
01AD    30                      TSX                     ; GET NUMBER FROM STACK
01AE    BD      0000G           JSR     FIX1
0181    EE      03              LDX     FIX1A,X         ; GET INTEGER NUMBER
0183    DF      00G             STX     R0,0
0185    30                      TSX
0186    BD      0000G           JSR     A9X
0189    35                      TXS
018A    5F                      CLR B
018B    96      00G             LDA A   ERRCD,0         ; TEST FOR ARGUMENT ERRORS
018D    26      3C              BNE     TYPCLR
018F    96      00G             LDA A   R0,0
01C1    26      38              BNE     TYPCLR
01C3    96      01G             LDA A   R0+1,0
01C5    27      05              BEQ     TYPMT           ; PASS CONTROL TO FILE SYSTEM IF NOT TYPE(0)
01C7    BD      0000G           JSR     TYPFIL
01CA    20      36              BRA     IP RTN
                                ;**********************************
                                ;       MAGTAPE TYPE FUNCTION
                                ;
01CC    5F              TYPMT:  CLR B
01C0    96      00G             LDA A   MTSTAT,0
01CF    85      00G             BIT A   MTOPEN,1                ; TEST IF FILE OPEN
01D1    27      28              BEQ     TYPCLR
01D3    85      00G             BIT A   MTSNEW,1
01D5    26      24              BNE     TYPCLR          ; IF NEW FILE THEN DONE
01D7    96      00G             LDA A   MTSTT2,0        ; TEST PRESENT BUFFER
01D9    85      00G             BIT A   MTSRD,1         ; SET IF BUFFER VALID
01DB    26      09              BNE     TYPMTT
01DD    BD      0000G           JSR     INITMT
01E0    BD      0000G           JSR     MTRBFR          ; GO GET A VALID BUFFER
01E3    BD      0000G           JSR     CRTRST
01E6    5F              TYPMTT: CLR B
01E7    96      00G             LDA A   MTSTAT,0        ; LOOK AT MAGTAPE FILE
01E9    85      00G             BIT A   MTSBIN,1
01EB    26      18              BNE     TYPBIN          ; BINARY
```

```
    58  01ED  85  00G             BIT A  MTSASC,I
    59  01EF  27  0A              BEQ    TYPCLR
    60  01F1  5C                  INC B
    61  01F2  DE  00G             LDX    MTPTR,D          ; ASCII
    62  01F4  A6  00              LDA A  0,X
    63  01F6  81  FF              CMP A  255.,I           ; EOF?
    64  01F8  27  01              BEQ    TYPCLR
    65  01FA  5C                  INC B
    66  01FB  37                  PSH B
    67  01FC  5F                  CLR B
    68  01FD  37                  PSH B
    69  01FE  37                  PSH B
    70  01FF  BD  0000G           JSR    FLOAT1           ; PUT VALUE BACK ON STACK
    71  0202  7E  0000G   IP.RTN: JMP    RTRN
    72  0205  C6  03      TYPBIN: LDA B  3,I              ; DO BINARY GOODIES
    73  0207  DE  00G             LDX    MTPTR,D
    74  0209  A6  00              LDA A  0,X
    75  020B  84  E0              AND A  224.,I           ; LOOK AT HEADER BITS
    76  020D  81  20              CMP A  32.,I
    77  020F  27  EA              BEQ    TYPCLR
    78  0211  5C                  INC B
    79  0212  81  40              CMP A  64.,I
    80  0214  27  E5              BEQ    TYPCLR
    81  0216  C6  01              LDA B  1,I
    82  0218  20  E1              BRA    TYPCLR
    83        0001'               .END
```

SYMBOL TABLE

```
ABRFLG= 0040        AFAIL = 0030        AMAT  = 0010        ARRAY = 0020        ASTR  = 0020
ATLOD = 0008        ATSNTG= XXXXXX G    ATWLD = 0004        A.END = XXXXXX G    A.MAX = XXXXXX G
A.PRIM= XXXXXX G    A.PTR = XXXXXX G    A.SEC = XXXXXX G    A.STAT= XXXXXX G    A.STRT= XXXXXX G
A7X   = XXXXXX G    A8X   = XXXXXX G    A9X   = XXXXXX G    BAKSTG= XXXXXX G    BANK  = XXXXXX G
BFRSTT= 0020        BINARY= 0001        BINBYT  0000R       BINCTL= 0000RG      BININ   0062R
BINITB  0069R       BINGT3  0138R       BINOUT  012CR       BLINK = XXXXXX G    BMAT  = 0004
BRKCNT= XXXXXX G    BSTR  = 0008        BUSACT= 0010        CDOPTR= XXXXXX G    CDSPTR= XXXXXX G
CHAR  = XXXXXX G    CHRCNT= XXXXXX G    CLPTR = XXXXXX G    CMAT  = 0001        COLCNT= XXXXXX G
CONCOD= XXXXXX G    CRDC3 = 0002        CREOF = 0008        CREOI = 0004        CREOT = 0020
CRETX = 0010        CRNORM= 0001        CRSTAT= XXXXXX G    CRTRST= XXXXXX G    CRVLD = 0080
CSTR  = 0002        CTKN  = XXXXXX G    CURSOR= XXXXXX G    DATDEV= 0022        DATEXT  018ER
DATIN   0148RG      DATOTH  0175R       DATSEC  0192R       DATSE.  0195R       DATSIR  0182R
DATVLC  017AR       DIMFLG= 0004        DIRCT = 0080        DISCNT= XXXXXX G    DISSRQ= 0008
DL    = XXXXXX G    DP    = XXXXXX G    DREXTA= XXXXXX G    DREXTB= XXXXXX G    DSPDEV= 0020
DT    = XXXXXX G    EDTBFR= XXXXXX G    ENDKEY= 0040        EOFTYP= 0038        EOLTG = XXXXXX G
EOSTG = XXXXXX G    ERATSN= XXXXXX G    ERDOMN= XXXXXX G    EREOM = XXXXXX G    ERFBFR= XXXXXX G
ERFILE= XXXXXX G    ERIOE = XXXXXX G    ERNLOD= XXXXXX G    ERNSEP= XXXXXX G    ERRCD = XXXXXX G
ERREAD= XXXXXX G    ERSTGL= XXXXXX G    ERTERM= XXXXXX G    ERUNDF= XXXXXX G    ESTG  = XXXXXX G
EXTFLG= 0080        FILDEV= 0C9C        FILEIN= XXXXXX G    FILEOT= XXXXXX G    FIX1  = XXXXXX G
FIX1A = 0003        FIX1B = 0004        FLOAT1= XXXXXX G    FNTWLD= 0008        FNFLG = 0010
FORTG = XXXXXX G    GLBFLG= XXXXXX G    GOSTG = XXXXXX G    GTDATA= XXXXXX G    IECIN = XXXXXX G
IECOUT= XXXXXX G    IMXTG = XXXXXX G    INITMT= XXXXXX G    INPUTK= 0001        IOBFR1= XXXXXX G
IOFLGS= XXXXXX G    IOFLNG= XXXXXX G    ITGCOD= 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    MTOPEN= XXXXXX G    MTPDEV= 0021
MTPD2 = 0023        MTPIN = XXXXXX G    MTPOUT= XXXXXX G    MTPTR = XXXXXX G    MTRBFR= XXXXXX G
MTSASC= XXXXXX G    MTSBIN= XXXXXX G    MTSNEW= XXXXXX G    MTSRD = XXXXXX G    MTSTAT= XXXXXX G
MTSTT2= XXXXXX G    NIODEV= XXXXXX G    NLPTR = XXXXXX G    NOKEY = 0080        NOOUT = XXXXXX G
NOWRIT= 0001        NTAPTR= 0008        NTATTR= 0004        NTDIMS= 0009        NTDLEN= 0005
NTLINK= 0000        NTNAME= 0002        NTPTR = XXXXXX G    NTRELV= 0010        NTSPTP= 0006
]]]]]
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    PNDEOF= XXXXXX G    PNDFLG= XXXXXX G    PNTNTG= XXXXXX G
PNTSTG= XXXXXX G    POINT = XXXXXX G    PPMODE= XXXXXX G    PRIDEF= 0001        PRTTG = XXXXXX G
PSCTG = XXXXXX G    PSHRET= XXXXXX G    PULFPN= XXXXXX G    READEOF 003CR       READHDR   000AR
READTG  0079RG      READVAL   0054RG    RECLEG= 0004        RESTRC  003FR       REXT      0053R
RFAIL = 00C0        RMAT  = 0040        RPTCTL= 0080        RSEXT   00E6R       RSLGS     0086R
RSSMS   0002R       RSTR  = 0080        RTRN  = XXXXXX G    RTRNTG= XXXXXX G    RUNFLG= 0080
RUNN  = 0002        RVERR     0048R     RVER      004AR     R0    = XXXXXX G    R1    = XXXXXX G
R10   = XXXXXX G    R11   = XXXXXX G    R12   = XXXXXX G    R13   = XXXXXX G    R14   = XXXXXX G
R15   = XXXXXX G    R16   = XXXXXX G    R17   = XXXXXX G    R18   = XXXXXX G    R19   = XXXXXX G
R2    = XXXXXX G    R20   = XXXXXX G    R21   = XXXXXX G    R22   = XXXXXX G    R23   = XXXXXX G
R3    = XXXXXX G    R4    = XXXXXX G    R5    = XXXXXX G    R6    = XXXXXX G    R7    = XXXXXX G
R8    = XXXXXX G    R9    = XXXXXX G    SBP   = XXXXXX G    SCALER= 0040        SCRTCH= XXXXXX G
SECDEF= 0002        SEMITG= XXXXXX G    SNDIT = 0080        SRTHDR= 0040        STAT37= XXXXXX G
STPFLG= 0040        STPKEY= 0020        STRING= 0010        SYSERR= XXXXXX G    TABPTR= XXXXXX G
TAGPTR= XXXXXX G    TCOL  = XXXXXX G    TP.RTN  0202R       TRCFLG= 0020        TYPBIN    0205R
TYPCLR  01F8R       TYPE      01AARG    TYPFIL= XXXXXX G    TYPMN     01CCR     TYPMTT    01E6R
T51DEV= 0025        UNDEF = 0080        VALTG = XXXXXX G    VALUND= 0040        VLUHDR    0190R
VULHDR= 2008        WRIERR    0114R     WRIPAT    0132R     WRISTG    00E7RG    WRIVAL    0119RG
WRIXXX  0118R       XAXIS = XXXXXX G    XFRCTL= XXXXXX G    YAXIS = XXXXXX G

  . ABS.   0000    00
           021A    01
```

```
BINCTL--BINARY READ,WRITE CONTR        RT-11 MMAC  VM02-10   14-OCT-76 01:28:42 PAGE 8+
SYMBOL TABLE

ERRORS DETECTED:  0      WARNINGS POSTED:  0      FREE CORE:  2383. WORDS
,SY:BINCTL/C(DK1:SEICLI,BINCTL
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
                1-279
A.END           1-196#  5-22*   5-28*   6-9*    6-14*
A.MAX           1-197#  2-11*   3-8*    4-25*   4-26*   4-43*   4-51*   4-54*   7-52
A.PRIM          1-192#  2-32    4-28
A.PTR           1-194#  2-10*   3-6*    4-20*   4-41*   4-49*   4-50    4-52    5-20*   6-7*    6-12*   6-19*   7-50    7-55*
A.SEC           1-193#
A.STAT          1-191#
A.STRT          1-195#
A7X             1-280#  3-7     6-13
A8X             1-283#
A9X             8-15#   8-28
ABRFLG          1-26#
ABAIL           1-155#
AMAT            1-154#
ARRAY           1-83#
AS12            1-153#
ATLOD           1-216#
ATSNTG          1-138#
A7VLD           1-204#
BAKSTG          1-146#
BANK            1-250#
BERSIT          1-201#
BINARY          1-295#  5-2     5-7     6-2     6-5
BINBYT          2-10#   2-17    2-24
BINCTL          1-278#  1-279#
BININ           2-12    3-13#   4-27    4-44    4-55
BINITB          3-13    3-18#
BINOTB          6-15    6-22#
BINOUT          5-23    6-10    6-15#   6-20
BLINK           1-222#
BMAT            1-157#
BRKCNT          1-73#
BSTR            1-156#
BUSACT          1-202#
CDOPTR          1-71#
CDSPTR          1-70#
CHAR            1-189#
CHRCNT          1-242#  4-15*
CLPTR           1-20#
CMAT            1-159#
COLCNT          1-244#
CONCOD          1-293#  7-26
CRDC7           1-181#
CREOF           1-183#
CREOI           1-182#
CREOT           1-185#
CRETX           1-184#
CRNORM          1-180#
CRSTAT          1-178#
CRTRST          8-14#   8-53
CRVLD           1-187#
CSTR            1-158#
CTKN            1-23#
CURSOR          1-223#
DATDEV          1-258#  4-29
DATEXT          7-9     7-12    7-41#
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

```
ITGCOD  7-2#    7-15
ITM1TG  1-136#
ITM2TG  1-137#
LITDEV  1-260#
JMPX    1-55#
KBDEV   1-255#
KBFLAG  1-210#
KBIN    1-208#
KEYFLG  1-35#
KEYSTK  1-52#
LBRKTG  1-139#
LCLFLG  1-24#
LDAX    1-65#
LDXX    1-60#
LENGTH  1-240#  4-5     4-8     4-9     4-10*   4-11*   4-31    4-35*   4-36    4-39*   4-45    4-47    5-10    5-11
        5-14    5-18*   5-19
LISTTG  1-127#
LNNOTG  1-135#
LOCTG   1-145#
LSP     1-46#
LSTFMT  1-227#
MTBFR   1-262#
MTOPEN  8-17#   8-44
MTPD2   1-259#  2-33
MTPDEV  1-257#
MTPIN   1-281#  3-24
MTPOUT  1-281#  6-28
MTPTR   1-291#  2-35    2-37*   8-20#   8-61    8-73
MTRBFR  8-16#   8-52
MTSASC  8-20#   8-58
MTSBIN  8-20#   8-56
MTSNEW  8-17#   8-46
MTSRD   8-18#   8-49
MTSTAT  8-20#   8-43    8-55
MTSTT2  8-18#   8-48
NIODEV  1-284#  3-20    3-21    3-22    3-25    6-24    6-25    6-26    6-27    6-29
NLPTR   1-21#
NOKEY   1-209#
NOOUT   1-225#
NOWRIT  1-226#
NTAPTR  1-93#
NTATTR  1-80#
NTDIMS  1-92#
NTDLEN  1-95#
NTLINK  1-77#
NTNAME  1-78#
NTPTR   1-22#
NTRELY  1-215#
NTSPTR  1-97#
NTVAL   1-87#
NTWCOL  1-91#
NTWLEN  1-96#
NTWROW  1-90#
NULLTG  1-122#
OBJATR  1-102#
OBJBCK  1-103#
OBJOT   1-104#
```

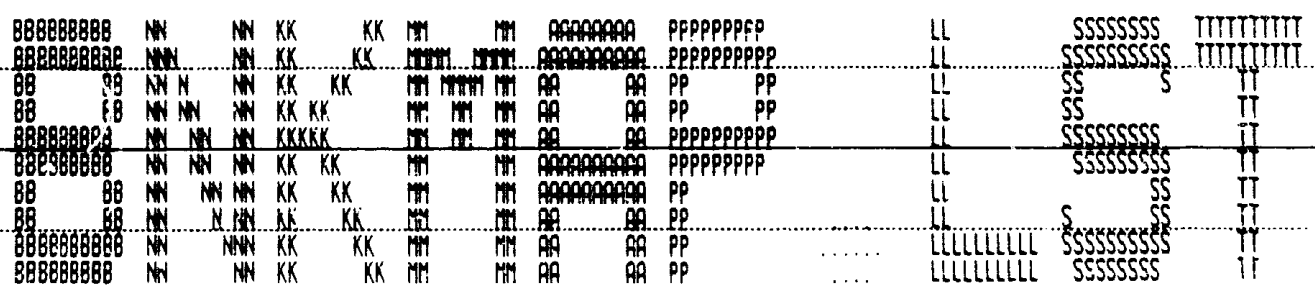
CROSS REFERENCE TABLE (CREF V01-03)

OBJLEN	1-101#													
ONSFLG	1-29#													
OPRADR	1-238#													
PAETG	1-133#													
PARM	1-85#													
PGMATR	1-109#													
PGMBP	1-111#													
PGMCD	1-113#													
PGMFP	1-110#													
PGMLEN	1-108#													
PGMLNN	1-112#													
PGMPTR	1-48#													
PGMTG	1-128#													
PLOSTG	1-127#													
PNDEOF	1-45#	2-18	2-38	2-40*										
PNDFLG	1-44#													
PNTNTG	1-132#													
PNTSTG	1-130#													
POINT	1-241#	3-5	4-19	4-21	4-22	5-8	5-9	5-25	6-11	6-18				
PPMODE	1-224#													
PRIDEF	1-206#													
PRITG	1-141#													
PSCTG	1-131#													
PSHRET	8-13#	8-22												
PULFPN	7-2#	7-24												
R0	1-38#	2-16	2-20	2-23	2-27	2-44	2-46*	4-6	4-7	4-13*	4-14	4-18*	4-23	4-24
	5-12*	5-13*	5-24	7-13	7-31*	7-34*	7-36*	8-26*	8-33	8-35				
R1	1-38#	7-29*	7-40*	7-46	7-49*									
R10	1-38#													
R11	1-38#													
R12	1-38#													
R13	1-38#													
R14	1-38#													
R15	1-38#													
R16	1-38#													
R17	1-39#													
R18	1-39#													
R19	1-39#													
R2	1-38#													
R20	1-39#													
R21	1-39#													
R22	1-39#													
R23	1-39#													
R3	1-38#	7-23												
R4	1-38#													
R5	1-38#													
R6	1-38#													
R7	1-38#													
R8	1-38#													
R9	1-38#													
REAEOF	2-19	2-25	2-41#											
REAHDR	2-16#	3-2	4-2											
REASTG	1-287#	4-2#												
REAVAL	1-289#	3-2#												
RECLFG	1-217#													
RESTRC	2-29	2-43#												
REXT	2-48	2-51	2-55#											

CROSS REFERENCE TABLE (CREF VM1-03)

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3%



```
                                        .TITLE  BNKMAP  GOODIES THAT ARE IN FIXED LOCATIONS
                                        .IDENT  /BNK024/
                                ;
                                ;       OVERLAY CONTROL VARIABLE
                                ;
                0000            OVER    =       0
                                ;
                0000                    .ASECT
                                ;
                                ;       BANK SWITCH CONTROL DATA
                                ;
                                ;       THIS IS DATA THAT GOES AT THE ORIGIN OF THE BANK SWITCH AREA
                                ;       FOR BANK ZERO.
                                ;
                                        .GLOBL  BNKADR,BNKEND,BNKMAP
                                ;
                                        .IF     EQ,OVER
                8800            BNKADR  =       H8800
                                        .IFF
                                BNKADR  =       H5000
                                        .ENDC
                8800                    .  = BNKADR
                8800            BNKMAP  =       .
                                ;
                                        .IF     EQ,OVER
        8800    4051                    .WORD   16465           ; BANK DATA IDENTIFER
        8802    0000                    .WORD   0               ; POWER UP ENTRY POINT
        8804    0000                    .WORD   0               ; INIT ENTRY POINT
        8806    0000                    .WORD   0               ; DELETE ALL ENTRY POINT
        8808    0000                    .WORD   0               ; PRE CLOSE ENTRY POINT
        880A    0000                    .WORD   0               ; TRANSFER VECTOR FOR SPECIAL ROM PACKS
        880C      00                    .BYTE   0               ; BANK ID (ROM PACK NUMBER)
                                ;
                                        .GLOBL  EXEC
        8800    45      58      45      .ASCII  /EXEC /
    8810        43      20      20
        8813    000006                  .WORD   EXEC
                                ;
        8815      00                    .BYTE   0               ; END OF TABLE MARKER
                                ;
                                        .ENDC
                                ;
                8816            BNKEND  =       .               ; TO LOOK AT IN LINK MAP
```

```
                        ;
                        ;       EXTENDED FUNCTION ROM BRANCH VECTOR LOCATION DEFINATIONS
                        ;
                        ;       TABLE FORMAT - ONE BYTE ENTRIES
                        ;       HIGH BYTE OF OPERATOR NAME MUST BE ZERO.
                        ;       LOW BYTE IS VVVV VOBB.
                        ;       VVVV VO IS THE VECTOR LOCATION
                        ;       BB IS THE FUNCTION BANK
                        ;
                                .MACRO  XFN     NAME,BANK
                                .GLOBL  NAME
                        NAME    =       VCTR*8.+BANK
                        VCTR    =       VCTR+1
                                .ENDM
                        ;
                0000    XFR     =       0               ;EXTENDED FUNCTION ROM
                0001    MTX     =       1
                0002    DSK     =       2               ;DISK I/O ROM
                0003    PGF     =       3               ;PAGE FULL ROM
                        ;
                0000    VCTR    =       0
        8816            XFN     AXIS,XFR
        8816            XFN     SCALE,XFR
        8816            XFN     VIEW,XFR
        8816            XFN     WINDOW,XFR
        8816            XFN     ROTATE,XFR
        8816            XFN     DRAW,XFR
        8816            XFN     RDRAW,XFR
        8816            XFN     MOVE,XFR
        8816            XFN     RMOVE,XFR
        8816            XFN     CROSS,XFR
        8816            XFN     GIN,XFR
        8816            XFN     ASC,XFR
        8816            XFN     CHR,XFR
        8816            XFN     LEN,XFR
        8816            XFN     POS,XFR
        8816            XFN     REP,XFR
        8816            XFN     SEG,XFR
        8816            XFN     CAT,XFR
        8816            XFN     VAL,XFR
        8816            XFN     STR,XFR
        8816            XFN     ASN,XFR
        8816            XFN     ACS,XFR
        8816            XFN     ATN,XFR
        8816            XFN     RND,XFR
        8816            XFN     SETFUZ,XFR

                0000    VCTR    =       0               ;MATRIX PACK
        8816            XFN     PGFVCT,PGF
        8816            XFN     FLUNIT,PGF
        8816            XFN     DET,MTX
        8816            XFN     INV,MTX
        8816            XFN     IRN,MTX
        8816            XFN     MPY,MTX
                        ;
                0001'           .END
```

```
BNKMAP GOODIES THAT ARE IN FIX RT-11 MMAC  VM02-10   14-OCT-76 01:28:58 PAGE 2+
SYMBOL TABLE

ACS    = 00A8 G      ASC    = 0058 G      ASN    = 00A0 G      ATN    = 0080 G      AXIS   = 0000 G
BNKADR= 8800 G      BNKEND= 8816 G      BNKMAP= 8800 G      CAT    = 0088 G      CHR    = 0060 G
CROSS  = 00C8 G      DET    = 0011 G      DRAW   = 0028 G      DSK    = 0002        EXEC   = ****** G
FLUNIT= 0008 G      GIN    = 0050 G      INV    = 0019 G      LEN    = 0068 G      MOVE   = 0038 G
MPY    = 0029 G      MTX    = 0001        OVER   = 0000        PGF    = 0003        PGF.VCT= 0003 G
POS    = 0070 G      RDRAW  = 0030 G      REP    = 0078 G      RMOVE  = 00A0 G      RND    = 0088 G
ROTATE= 0020 G      SCALE  = 0008 G      SEG    = 0080 G      SETFUZ= 00C0 G      STR    = 0098 G
TRN    = 0021 G      VAL    = 0090 G      VCTR   = 0006        VIEW   = 0010 G      WINDOW= 0018 G
XFR    = 0000
. ABS.    8816    00
          0000    01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3253.  WORDS
.SY:BNKMAP/C(DK1:SEICLI,BNKMAP
```

CROSS REFERENCE TABLE (CREF V01-03.)

	1-37#	1-38	1-57											
ACS	2-43#	2-43#												
ASC	2-33#	2-33#												
ASN	2-42#	2-42#												
ATN	2-44#	2-44#												
AXIS	2-22#	2-22#												
BNKADR	1-30#	1-37#	1-37											
BNKEND	1-30#	1-57#												
BNKMAP	1-30#	1-38#												
CAT	2-39#	2-39#												
CHR	2-34#	2-34#												
CROSS	2-31#	2-31#												
DET	2-51#	2-51#												
DRAW	2-27#	2-27#												
DSK	2-18#													
EXEC	1-49#	1-51												
FLUNIT	2-50#	2-50#												
GIN	2-32#	2-32#												
INV	2-52#	2-52#												
LEN	2-35#	2-35#												
MOVE	2-29#	2-29#												
MPY	2-54#	2-54#												
MTX	2-17#	2-51	2-52	2-53	2-54									
OVER	1-21#	1-32	1-40											
PGF	2-19#	2-49	2-50											
PGFVCT	2-49#	2-49#												
POS	2-36#	2-36#												
RDRAW	2-28#	2-28#												
REP	2-37#	2-37#												
RMOVE	2-30#	2-30#												
RND	2-45#	2-45#												
ROTATE	2-26#	2-26#												
SCALE	2-23#	2-23#												
SEG	2-38#	2-38#												
SETFUZ	2-46#	2-46#												
STR	2-41#	2-41#												
TRN	2-53#	2-53#												
VAL	2-40#	2-40#												
VCTR	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#	2-39	2-39	2-39#	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#	2-48#	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#			
VIEW	2-24#	2-24#												
WINDOW	2-25#	2-25#												
XFR	2-16#	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		

CROSS REFERENCE TABLE (CREF V01-03.)

SEI 1-38

XFN 2-10# 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 2-49 2-50
2-51 2-52 2-53 2-54

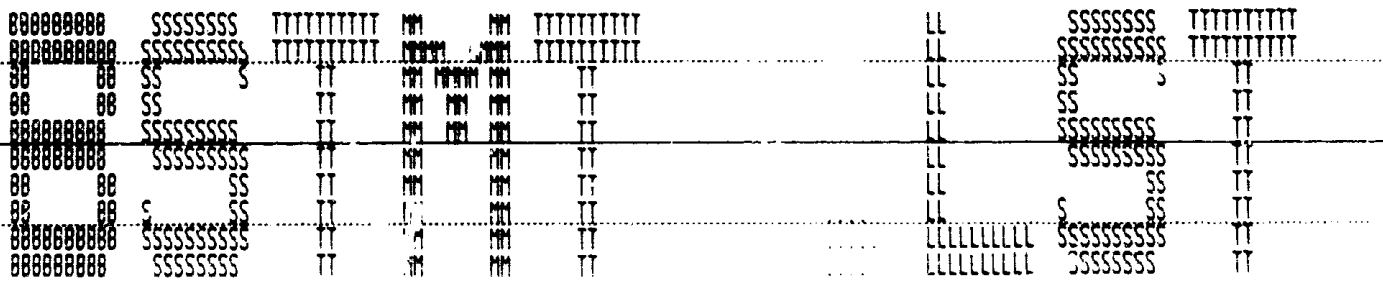


TABLE OF CONTENTS

```
                                        .TITLE  BSTMT   ASSORTED BASIC STATEMENTS
                                        .IDENT  /BE4043/
                                ;
                                        .GLOBL  BSTMT
0000'                           BSTMT   =.
                                        .GLOBL  IOPROC,PRINT,FIX1,FLOAT1,INTMLC,COMPR
                                        .GLOBL  FUDGH,FUDGL
                                        .GLOBL  GETLNA,R9X,ISTINT
                                        .GLOBL  USRORG,XAXIS,ZX
                                        .GLOBL  BNKADR,RUNBNK
                                        .GLOBL  PSHFPN,FPADD,FPZERO
                                        .GLOBL  SETERR,SETERA,HALTR,HALTA
                                ;
                                        .SBTTL  XXX EOF         END OF FILE TOKEN IN ON/OFF STMT
                                        .GLOBL  EOF
                                ;
                                ;       SET UP OR DELETE LINKAGE FOR EOF ON UNITS
                                ;       LOGICAL UNIT NO IS ON STACK.
                                ;       NUMBER IS LEFT ON STACK.
                                ;
0000    30                      EOF:    TSX                     ;SET UP FOR FIX
0001    BD      0000G                   JSR     FIX1
0004    26      48                      BNE     EOFERR
0006    30                              TSX                     ;TEST UNIT NO.
0007    A6      03                      LDA A   3,X             ;FIRST BYTE MUST BE ZERO
0009    26      43                      BNE     EOFERR
                                ;*******************************************************************
                                ;       LDA B   5,X             ;ONE TO NINE
0008    E6      04                      LDA B   4,X             ;GET THE CORRECT SECOND BYTE
                                ;*******************************************************************
000D    C1      09                      CMP B   9.,1
000F    22      30                      BHI     EOFERR
0011    CB      08                      ADD B   8.,1            ;ADD BIAS FOR ON UNITS
0013    4F                      ONDEF:  CLR A
0014    58                              ASL B                   ;ON UNIT TABLE IS 2 BYTES WIDE
0015    F8      0053'                   ADD B   AONTBL+1        ;ADD BASE ADDR FOR TBL
0018    B9      0052'                   ADC A   AONTBL
001B    97      00G                     STA A   R7,D            ;TRAN A & B TO X
001D    D7      01G                     STA B   R7+1,D
001F    DE      00G                     LDX     R7,D
0021    96      00G                     LDA A   LCLFLG,D        ;IN ON STMT AT DEFINITION TIME
0023    85      02                      BIT A   ONSFLG,I
0025    26      0A                      BNE     ONSTMT          ;ON STMT ACTIVE
0027                                    SEI
0027    01      0F              .BYTE   01,17
0029    6F      00                      CLR     0,X             ;OFF STMT - RESET TBL ENTRY
002B    6F      01                      CLR     1,X
002D    0E                              CLI
002E    7E      0000G                   JMP     DREXTA
                                ;
0031    DE      00G             ONSTMT: LDX     NTPTR,D         ;PUSH STMT NO ON TO STACK
0033    EE      01                      LDX     1,X
0035    DF      00G                     STX     R0,D
0037    BD      0000G                   JSR     GETLNA          ;CONVERT TO LINE ADDR
003A    96      00G                     LDA A   ERRCD,D         ;DID IT WORK
003C    26      0D                      BNE     ONEXT           ;NO
003E    DE      00G                     LDX     R7,D
```

```
            0040                            SEI
      0040   01      0E             .BYTE   01,17
            0042     96    00G              LDA A   RO,D          ;MOVE CLPTR TO DEFINE ON UNIT
            0044     A7    00               STA A   0,X
            0046     96    01G              LDA A   RO+1,D
            0048     A7    01               STA A   1,X
            004A     0E                     CLI
            004B     7E    0000G    ONEXT:  JMP     HALTA
                                    ;
            004E     BD    0000G    EOFERR: JSR     SETERA
            0051     00G                    .BYTE   EREOFN
                                    ;
            0052   0000G            AONTBL: .WORD   ONTBL
```

```
                                        .SBTTL  XXX ONUNIT      ON UNIT TOKENS
                                        .GLOBL  ONUNIT
                                ;
                                ;       SET UP OR DELETE LINKAGE FOR ON UNITS EXCEPT EOF.
                                ;       THIS JOINS CODE IN ABOVE ROUTINES
                                ;
0054    D6      00G     ONUNIT: LDA B   CTKN,D          ;GET TOKEN FOR UNIT INDEX
0056    C0      00G             SUB B   SIZCOD,I                ;MUST BE FIRST ON UNIT TOKEN
0058    5C                      INC B
0059    20      88              BRA     ONDEF           ;JOIN COMMON CODE
                                ;
                                ;
                                        .SBTTL  XXX OFF         OFF STATEMENT
                                        .GLOBL  OFF
                                ;
                                ;       IF BARE OFF STATEMENT DELETE ALL ON UNITS.
                                ;       IF THERE ARE TOKENS IN OFF LINE THIS IS A NOP.
                                ;
005B    DE      00G     OFF:    LDX     NTPTR,D         ;IF OFF WITH NO UNITS CLEAR THEM ALL
005D    A6      00              LDA A   0,X
005F    81      00G             CMP A   CRCOD,I
0061    26      0B              BNE     OFFEXT
0063    CE      0000G           LDX     ONTBL,I         ;CLEAR ON TABLE
0066    86      1E              LDA A   30.,I           ;BYTES TO CLEAR
0068    6F      00      OFFLP:  CLR     0,X
006A    08                      INX
006B    4A                      DEC A
006C    26      FA              BNE     OFFLP
006E    39              OFFEXT: RTS
```

```
                                        .SBTTL  XXX ON          ON UNIT DEFINITION STATEMENT
                                        .GLOBL  ON
                                ;
                                ;       SET UP FLAG FOR ON/OFF UNIT ROUTINES.
                                ;
006F    96      00G     ON:     LDA A   GLBFLG,D
0071    85      C0              BIT A   192.,I
0073    27      07              BEQ     ONERR           ;IF IN IMEDIATE MODE SET ERROR
0075    96      00G             LDA A   LCLFLG,D
0077    8A      02              ORA A   ONSFLG,I
0079    97      00G             STA A   LCLFLG,D
007B    39                      RTS
                                ;
007C    BD      0000G   ONERR:  JSR     SETERR
007F    00G                     .BYTE   ERNIMX
```

```
                                        .SBTTL  XXX EOL         END OF LINE ROUTINE
                                        .GLOBL  EOL
                                ;
                                ;       SET UP TO ABORT LINE AND IF IN IMMEDIATE EXEC MODE PRINT JUNK ON STACK
                                ;
0080    BD      0000G   EOL:    JSR     HALTR           ;SET ABORT LINE
0083    96      00G             LDA A   GLBFLG,D        ;SEE IF I AM IN RUN MODE
0085    2A      01              BPL     EOLPRT          ;GO TO PRINT (IOPROC) IF IN CALC MODE
0087    39              EOLRTS: RTS
                                ;
0088    30              EOLPRT: TSX                     ;IF STACK IS EMPTY EXIT
0089    A6      02              LDA A   2,X             ;GET TAG
008B    81      00G             CMP A   EOLTG.I
008D    27      F8              BEQ     EOLRTS          ;GO HOME IF EOL
008F    CE      0000G           LDX     PRINT,I         ;SET UP TO CALL PRINT
0092    DF      00G             STX     OPRADR,D
0094    7E      0000G           JMP     IOPROC
```

```
                                        .SBTTL  *** INIT        INIT ROUTINE - SET SYSTEM DEFULTS
                                        .GLOBL  INIT,INITX,SRQOFF
                                ;
                                ;       SET UP SYSTEM DEFULTS FOR THE USER
                                ;
0097  CE  0004G       INIT:     LDX     BNKADR+4,I      ;INIT ENTRY FOR ROM PACKS
009A  86  00G         INITX:    LDA A   EOLTG,I         ;FAKE AN END OF LINE FOR CLOSE
009C  36                        PSH A
009D  BD  0000G                 JSR     RUNBNK          ;GO CALL ALL BANKS
00A0  BD  0175'                 JSR     RAD             ;RADIANS FOR TRIG
                                .GLOBL  SETRND
00A3  BD  0000G                 JSR     SETRND
00A6  CE  0010G                 LDX     NGTAB+16.,I     ;SET UP FOZZ
00A9  BD  0000G                 JSR     PSHFPN
00AC  CE  0000G                 LDX     FUZB,I
00AF  ED  0000G                 JSR     PULFPN
00B2  86  28                    LDA A   40.,I
00B4  B7  0000G                 STA A   FUZA
                                .GLOBL  NGTAB,FUZA,FUZB,PULFPN,PSHFPN
00B7  DE  00G                   LDX     STPTR,D         ;SET UP VARIABLE AS UNDEFINDE
00B9  A6  04          INTLPA:   LDA A   NTATTR,X        ;GET ATTR BYTE
00BB  84  10                    AND A   STRING,I        ;STRING BIT IS THE ONLY ONE TOSAVE
00BD  26  02                    BNE     INTSKP          ;IF THIS IS A STRING SKIP
00BF  8A  40                    ORA A   SCALER,I        ;IF NOT MAKE IT SCALER
00C1  8A  80          INTSKP:   ORA A   UNDEF,I         ;ALL VARS ARE UNDEFINED
00C3  A7  04                    STA A   NTATTR,X
00C5  EE  00                    LDX     NTLINK,X        ;TRY NEXT ENTRY
00C7  26  F0                    BNE     INTLPA
00C9  07                        TPA                     ;SAVE INTERRUPT STATUS
00CA                            SEI
00CA  01                .BYTE   01,17
00CC  CE  0000G                 LDX     CNTBL,I
00CF  6F  00          INTLPB:   CLR     0,X
00D1  08                        INX
                                .GLOBL  ENDTBL
00D2  8C  0000G                 CPX     ENDTBL,I
00D5  26  F8                    BNE     INTLPB
00D7  D6  00G                   LDA B   GLBFLG,D
00D9  C4  D6                    AND B   255.-TRCFLG-DISSRQ-NCSFLG,I
00DB  CA  10                    ORA B   KEYFLG,I
00DD  D7  00G                   STA B   GLBFLG,D
00DF  7F  0000G                 CLR     PNDFLG
00E2  06                        TAP                     ;RESET INTERRUPT MASK
                                .GLOBL  STAT37
00E3  7F  0000G                 CLR     STAT37
                                .GLOBL  RESTZ
00E6  BD  0000G                 JSR     RESTZ           ;RESET DATA STATEMENT POINTERS
                                .GLOBL  CLOSE
00E9  BD  C000G                 JSR     CLOSE
                                .GLOBL  IECIFC
00EC  BD  0000G                 JSR     IECIFC
00EF  7F  0000G                 CLR     ERRCD
00F2  32                        PUL A
00F3  96  00G         SRQOFF:   LDA A   PNDFLG,D                ;RESET SRQ PENDING STATE
00F5  84  EF                    AND A   239.,I
00F7  97  00G                   STA A   PNDFLG,D
00F9  39                        RTS
```

```
                                                .SBTTL  *** DLTALL      DELETE ALL FUNCTION
                                                .GLOBL  DLTALL
                                        ;
                                        ;       DO AN INIT AND CLEAN UP STACKS
                                        ;
00FA                                    DLTALL: SEI                     ;DISABLE INTERRUPT MASK
00FA    01      CF                      .BYTE   01.17
00FC                                    PLRA                            ;SAVE RETURN ADDR FOR LATER
00FC    32                              PUL A
00FD    97      01G                     STA A   DREXTA+1,D
00FF    32                              PUL A
0100    97      02G                     STA A   DREXTA+2,D
0102    DE      00G                     LDX     USRORG,D        ;RESET LOW STACK
0104    DF      00G                     STX     LSP,D
0106    DE      00G                     LDX     STPTR,D         ;GET SYMBOL TABLE POINTER
0108    6F      00                      CLR     0,X             ;RESET LINK IN PARM ENTRY
010A    6F      01                      CLR     1,X
010C    09                              DEX
010D    DF      00G                     STX     SBP,D           ;NEW HIGH END OF EXECUTION STACK
010F    9E      00G                     LDS     SBP,D           ;RESET THE MAIN STACK
0111    86      00G                     LDA A   EOSTG,I         ;MARK IT
0113    36                              PSH A
0114    DE      00G                     LDX     ZX,D            ;RESET PGMPTR
0116    DF      00G                     STX     PGMPTR,D
0118    DF      00G                     STX     CLPTR,D
011A    DF      00G                     STX     NLPTR,D
011C    7F      00000G                  CLR     XAXIS
011F    0E                              CLI                     ;RESET MASK
0120    CE      0006G                   LDX     #NADR+6,I       ;DELETE ALL ENTRY ADDR
0123    BD      00'4C'                  JSR     INITX
0126    7E      0J00G                   JMP     DREXTA          ;GO TO SAVED RETURN ADDR
```

```
                                        .SBTTL  XXX SPACE/MEMORY FUNCTIONS
                                        .GLOBL  SPACE,MEMORY
                                ;
                                ;       ROUTINE TO IMPLEMENT BASIC 'SPACE' FUNCTION
                                ;
0129  CE  0000G   SPACE: LDX     FPZERO,I
012C  BD  0000G          JSR     PSHFPN
012F  DE  00G            LDX     PGMPTR,D        ;GET ADDR OF FIRST LINE
0131  27  11             BEQ     SPCSKP          ;TRA IF NULL PROGRAM
0133  DF  00G    SPCLP: STX     R0,D
0135  CE  0147'          LDX     FP72,I
0138  BD  0000G          JSR     PSHFPN
013B  BD  0000G          JSR     FPADD
013E  DE  00G            LDX     R0,D
0140  EE  03             LDX     PGMFP,X
0142  26  EF             BNE     SPCLP
0144  7E  0000G  SPCSKP: JMP     DREXTA
                                ;
0147  04  07  90 FP72:   .BYTE   4,7,144.,0,0,0,0,0
014A  00  00  00
014D  00  00
                                ;
                                ;       ROUTINE TO IMEPLEMENT THE MEMORY FUNCTION
                                ;
014F  4F         MEMORY: CLR A
0150  36                 PSH A
0151  36                 PSH A
0152  86  00G            LDA A   ITMITG,I
0154  36                 PSH A
0155  BD  0000G          JSR     COMPR
0158  9F  00G            STS     R0,D            ;
015A  96  00G            LDA A   R0,D            ;FIND SPACE
015C  D6  01G            LDA B   R0+1,D
015E  D0  01G            SUB A   LSP+1,D
0160  92  00G            SBC A   LSP,D
0162  C0  00G            SUB B   FUDGL,I
0164  82  00G            SBC A   FUDGH,I
0166  2A  02             BPL     COMMON
0168  4F                 CLR A
0169  5F                 CLR B
016A  37         COMMON: PSH B
016B  36                 PSH A
016C  86  00G            LDA A   ITMITG,I
016E  36                 PSH A
016F  BD  0000G          JSR     FLOAT1
0172  7E  0000G          JMP     DREXTA
```

```
                                        .SBTTL  *** RAD/DEG/GRAD
                                        .GLOBL  RAD,DEG,GRAD
                                ;
                                ;       ROUTINES TO IMPLEMENT THE RAD,DEG,GRAD PORTIONS OF THE SET COMMAND.
                                ;
                                        .GLOBL  CONV1,RADCON,DEGCON,GRDCON
                                ;
0175    CE      0000G   RAD:    LDX     #RADCON,1
0178    FF      0000G   ANGLE:  STX     CONV1
017B    39                      RTS
                        ;
017C    CE      0000G   DEG:    LDX     #DEGCON,1
017F    20      F7              BRA     ANGLE
                        ;
0181    CE      0000G   GRAD:   LDX     #GRDCON,1
0184    20      F2              BRA     ANGLE
```

```
                                        .SBTTL  XXX SET COMMANDS
                                        .GLOBL  TRACE,NORMAL,KEY,NOKEY,CASE,NOCASE
                                ;
                                ;       ROUTINES TO IMPLEMENT TRACE, NORMAL, KEY, NOKEY, CASE AND NOCASE
                                ;       PORTIONS OF SET COMMAND.
                                ;
                                ;       SET UP SYSTEM OPTIONS
                                ;
0186  86  20      TRACE:  LDA A   TRCFLG,I
0188  9A  00G     ORGLB:  ORA A   GLBFLG,D
018A  97  00G             STA A   GLBFLG,D
018C  39                  RTS
                                ;
018D  ..  DF      NORMAL: LDA A   255.-TRCFLG,I
018F  94  00G     ANDGLB: AND A   GLBFLG,D
0191  97  00G             STA A   GLBFLG,D
0193  39                  RTS
                                ;
0194  86  EF      KEY:    LDA A   255.-KEYFLG,I
0196  20  E7              BRA     ANDGLB
                                ;
0198  86  10      NOKEY:  LD  A   KEYFLG,I
019A  20  EC              BRA     ORGLB
                                ;
019C  86  FE      CASE:   LDA A   255.-NCSFLG,I
019E  20  EF              BRA     ANDGLB
                                ;
01A0  86  01      NOCASE: LDA A   NCSFLG,I
01A2  20  E4              BRA     ORGLB
```

```
                                        .SBTTL  XXX WAIT        WAIT FOR INTERRUPT COMMAND
                                        .GLOBL  WAIT
                                ;
                                ;       ROUTINE TO IMPLEMENT THE WAIT COMMAND
                                ;
                                ;       USES BIT (OUAFLG) SET BY TSTINT IF A ON UNIT WAS RUN IN THE LAST
                                ;       CALL TO CONTROL BUSY LOOP.
                                ;
01A4    86      000     WAIT:   LDA A   RTRNTG,I        ;TAG RETURN ADDRESS
01A6    36                      PSH A
01A7    96      000             LDA A   GLBFLG,D
01A9    84      FD              AND A   255.-OUAFLG,I   ;RESET BIT
01AB    97      000             STA A   GLBFLG,D
01AD    BD      00000G  WAITLP: JSR     TSTINT          ;GO TRY FOR ONE
0180    96      000             LDA A   GLBFLG,D        ;SEE IF ON UNIT RAN
0182    85      02              BIT A   OUAFLG,I
0184    27      F7              BEQ     WAITLP          ;IF NOT TRY AGAIN
0186    32                      PUL A
0187    39                      RTS
                                ;
        0001'                   .END
```

```
BSTMT  ASSORTED BASIC STATEMEN RT-11 MMAC  VM02-10   14-OCT-76 01:25:04 PAGE 16+
SYMBOL TABLE

ABRFLG=  0040        AFAIL =  0030        ALLOK =  0004        ALLTG = xxxxxx G     AMAT  =  0010
ANDGLB   018ER       ANGLE    0178R       ANNTBL   0062R       ARRAY =  0020        ASGCOD= xxxxxx G
ASTR  =  0020        ATSNTG= xxxxxx G     A9X   = xxxxxx G     BAKSTG= xxxxxx G     BMAT  =  0004
BNKADR= xxxxxx G     BSTMT :  0000RG      BSTR  =  0008        CALLTG= xxxxxx G     CASE     019CRG
CDOPTR= xxxxxx G     COSPTR= xxxxxx G     CLOSE = xxxxxx G     CLPTR = xxxxxx G     CMAT  =  0001
COMMON   016AR       COMPR = xxxxxx G     CONCOD= xxxxxx G     CONV1 = xxxxxx G     CRCOD = xxxxxx G
CSTR  =  0002        CTKN  = xxxxxx G     DATCOD= xxxxxx G     DEG      017CRG     DEGCON= xxxxxx G
DIMFLG=  0040        DISSRC=  0008        DLTALL   00EARG      DREXTA= xxxxxx G     DREXTB= xxxxxx G
ENDTBL= xxxxxx G     EOF      0000RG      EOFCOD= xxxxxx G     EOFERR   004ER       EOFTBL= xxxxxx G
EOL      0080RG      EOLPRT   0088R       EOLRTS   0087R       EOLTG = xxxxxx G     EOSTG = xxxxxx G
EQUCOD= xxxxxx G     ERRSGN= xxxxxx G     ERRBRK = xxxxxx G     ERRDMN= 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
ERSHOP= xxxxxx G     ERSTOP= xxxxxx G     ERUNDF= xxxxxx G     ERVAL = xxxxxx G     ERWSFL= xxxxxx G
ESTG  = xxxxxx G     EXTFLG=  0080        FIX1  = xxxxxx G     FLOAT1= xxxxxx G     FNACOD= xxxxxx G
FNFLG =  0010        FNTBL = xxxxxx G     FORTG = xxxxxx G     FPADD = xxxxxx G     FPZERO= xxxxxx G
FPZ2     0147R       FUDGH = xxxxxx G     FUDGL = xxxxxx G     FUZA  = xxxxxx G     FUZB  = xxxxxx G
GETLNA= xxxxxx G     GLBFLG= xxxxxx G     GOSTG = xxxxxx G     GRAD     0181RG     GRDCON= xxxxxx G
HALTA = xxxxxx G     HALTR = xxxxxx G     IECIFC= xxxxxx G     IMACOD= xxxxxx G     IMXFLG=  0020
INXTG = xxxxxx G     INIT     0097RG      INITX    0090RG      INTLPA   0089R       INTLPB   00CFR
INTMLC= xxxxxx G     INTSKP   00C1R       IOPROC= xxxxxx G     ITM1TG= xxxxxx G     ITM2TG= xxxxxx G
JMPAX = xxxxxx G     JMPX  = xxxxxx G     KEY      0134RG      KEYFLG=  0010        KEYSTK= xxxxxx G
LBRKTG= xxxxxx G     LCLFLG= xxxxxx G     LDBX  = xxxxxx G     LDBX  = xxxxxx G     LDXX  = xxxxxx G
LISTTG= xxxxxx G     LITCOD= xxxxxx G     LNNOTG= xxxxxx G     LSP   = xxxxxx G     LSTCOD= xxxxxx G
MEMORY   014FRG      MMICOD= xxxxxx G     MPLCOD= xxxxxx G     MULCOD= xxxxxx G     NCSFLG=  0001
NGTBR = xxxxxx G     NLPTR = xxxxxx G     NOCASE   0180RG      NOKEY    0198RG      NORMAL   0180RG
NTAPTR=  0008        NTATTR=  0004        NTDIMS=  0009        NTDLEN=  0005        NTLINK=  0000
NTNAME=  0002        NTPTR = xxxxxx G     NTS PTR=  0008       NTVAL =  0005        NTWCOL=  0007
NTXLEN=  0007        NTXROW=  0006        NULLTG= xxxxxx G     OBJATR=  0002        OBJBCK=  0003
OBJDT =  0005        OBJLEN=  0000        OFF      0058RG      OFFEXT   006ER       OFFLP    0068R
ON       006FRG      ONDEF    0013R       ONERR    007CR       ONEXT    004BR       ONSFLG=  0002
ONSTMT   0031R       ONTBL = xxxxxx G     ONUNIT   0054RG      OPRADR= xxxxxx G     ORGLB    0188R
OUAFLG=  0002        PAETG = xxxxxx G     PARM  =  0008        PGMATR=  0002        PGMBP =  0006
PGMCD =  0009        PGMFP =  0003        PGMLEN=  0000        PGMLNM=  0007        PGMPTR= xxxxxx G
PGMTG = xxxxxx G     PLOSTG= xxxxxx G     PNDECF= xxxxxx G     PNDFLG= xxxxxx G     PNTNTG= xxxxxx G
PNTSTG= xxxxxx G     PRINT = xxxxxx G     PRTTG = xxxxxx G     PSCTG = xxxxxx G     PSHFPN= xxxxxx G
PULFPN= xxxxxx G     RAD      0175RG      RADCON= xxxxxx G     RESTZ = xxxxxx G     RFAIL =  00C0
RMAT  =  0040        RSTR  =  0080        RTRNTG= xxxxxx G     RUNBNK= 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
SETERA= xxxxxx G     SETERR= xxxxxx G     SETRND= xxxxxx G     SIZCOD= xxxxxx G     SPACE    0129RG
SPCLP    0133R       SPCSKP   0144R       SRGOFF   00F3RG      STAT37= xxxxxx G     STAX  = xxxxxx G
STPFLG=  0040        STPTR = xxxxxx G     STRING=  0010        TRACE    0186RG      TRCFLG=  0020
TSTINT= xxxxxx G     UNDEF =  0080        USRORG= xxxxxx G     VALERR=  0040        VALTG = xxxxxx G
WAIT     01A4RG      WAITLP   01ADR       XAXIS = xxxxxx G     ZX    = xxxxxx G
  ABS.   0000     (0)
         0188     01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2446. WORDS
.SY:BSTMT/C(OK):SE(CL),BSTMT
```

CROSS REFERENCE TABLE (CREF V01-03)

```
        7-5
A9X     7-8#
ABRFLG  2-10#
ABSIL   5-38#
ALLOK   4-13#
ALLTG   5-24#
AMAT    5-37#
ANDGLB  15-15#  15-20   15-26
ANGLE   14-9#   14-13   14-16
AONTBL  7-36    7-37    7-68#
ARRAY   4-10#
ASGCOD  1-36#
ASTR    5-36#
ATSNTG  5-22#
BAKSTG  5-23#
BMAT    5-40#
BNKADR  7-10#   11-6    12-24
BSTMT   7-4#    7-5#
BSTR    5-39#
CALLTG  5-27#
CASE    15-2#   15-25#
CDOPTR  3-40#
CDSPTR  3-39#
CLOSE   11-47#  11-48
CLPTR   2-4#    12-20*
CMAT    5-42#
COMMON  13-37   13-40#
COMPR   7-6#    13-29
CONCOD  1-45#
CONV1   14-6#   14-9*
CRCOD   1-44#   8-21
CSTR    5-41#
CTKN    2-7#    8-7
DATCOD  1-47#
DEG     14-2#   14-12#
DEGCON  14-6#   14-12
DIMFLG  2-13#
DISSRQ  2-21#   11-38
DLTALL  12-2#   12-6#
DREXTA  2-25#   7-48    12-7*   12-7*   12-26   13-17   13-45
DREXTB  2-26#
ENDTBL  11-34#  11-35
EOF     7-15#   7-21#
EOFCOD  1-41#
EOFERR  7-23    7-26    7-32    7-65#
EOFTBL  3-5#
EOL     10-2#   10-6#
EOLPRT  10-8    10-11#
EOLRTS  10-9#   10-14
EQLTG   5-28#   10-13   11-7
EQSTG   5-29#   12-16
EQUCOD  1-37#
ERASGN  6-16#
ERBRK   6-14#
ERDOMN  6-7#
EREOFN  6-15#   7-66
```

CROSS REFERENCE TABLE (CREF.MO1-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03)

PLRA 1-20# 12-7
PLRB 1-27#
SE1 1-3# 7-44 7-57 11-30 12-6

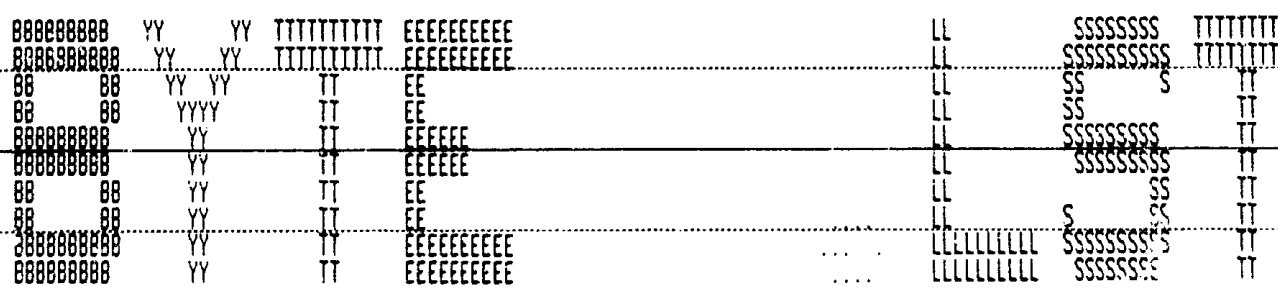


TABLE OF CONTENTS

```
                                        .TITLE  BYTE--RBYTE, WBYTE I
                                        .SBTTL  READ BYTE, WRITE BYTE ROUTINES
                                        .IDENT  /SBWO10/


                                        .GLOBL  ATNON, ATNOFF, UNADR, INTSEC, PIALT
                                        .GLOBL  SPC10B, IOCLNR  LOCTG, INTACP
                                        .GLOBL  RBYTE, WBYTE
                                        .GLOBL  A9X, PSHFPW, FIYI, ERFIXN

                                    ;THIS ROUTINE SETS UP THE CONTROL FOR THE WBYTE COMMAND AND PERFORMS
                                    ;SAID FUNCTION ON THE IEC BUS.

0000    86      00G         ;WBYTE: LDA A   EOLTG.I         ;TAG THIS
0002    36                          PSH A
0003    9F      00G                 STS     R0,D
                            ;TURN ON I/O LIGHT
0005    B6      0000G               LDA A   PIALT
0008    8A      40                  ORA A   64.,I
000B    B7      0000G               STA A   PIALT
                            ;
0000    86      11                  LDA A   17.,I
000E    97      00G                 STA A   IOFUNC,D
0011    96      00G                 LDA A   A.STAT,D        ; SEE IF  ON STACK
0013    85      04                  BIT A   ATVLD,I
0015    27      13                  BEQ     WBMATN
                            ;
0017    BD      0000G               JSR     ATNON
001A    86      00G                 LDA A   ATSNTG.I        ;GO FOR  INFO
001C    BD      0000G               JSR     LOCTG
001F    96      00G                 LDA A   PPMODE,D        ; LOOK FOR   MODE FORMAT
0021    97      00G                 STA A   R20,D           ; SAVE FACT FOR LATER
0023    BD      0000G               JSR     SPC10B          ;GO OUTPUT THIS STUFF.
0026    96      00G                 LDA A   R20,D           ; SEE IF WBY N: FORMAT
0028    26      05                  BNE     WBEXT
002A    9F      00G         WBMATN: STS     R0,D
002C    BD      0000G               JSR     SPC100          ;OUTPUT WITHOUT ATTN
002F    BD      0000G       WBEXT:  JSR     IOCLNR
                            ;THIS IS PART 1 OF A 3 PART PATCH TO FIX WBY N: FORMAT
                            ;THIS LOCATION IS JUMPED TO OUT OF UNADR UPON CLEAN UP OF ANY
                            ;I/O STATEMENT
                            ;
                                        .GLOBL  WBYP1
0032    81      11          WBYP1:  CMP A   17.,I           ; IS IT WBYTE?
0034    27      21                  BEQ     WBYP2           ; IF SO GO TO SECOND PART OF PATCH
0036    39                          RTS                     ; ELSE RETURN
```

```
                                        .SBTTL  RBYTE--READ BYTE CONTROLLER
                                ;THIS ROUTINE SETS UP THE IO FOR THE RBYTE TRANSFER AND DOES IT
0037    86      00G     RBYTE:  LDA A   EOLTG,I         ;TAG IT.
0039    36              00G             PSH A
003A    9F              00G             STS     RQ.D
                                ;TURN ON THE I/O LIGHT
003C    B6      0000G           LDA A   PIALT
003F    8A      40              ORA A   64.,I
0041    B7      0000G           STA A   PIALT
                                ;
0044    BD      0000G           JSR     INIACP          ;SET UP BUS
0047    86      12              LDA A   18.,I
0049    97      00G             STA A   IOFUNC,D
004B    96      00G             LDA A   A.STAT,D
004D    8A      80              ORA A   DIRCT,I
004F    97      00G             STA A   A.STAT,D
0051    BD      0000G           JSR     SPCIOB          ;GO INPUT THIS MESS
0054    BD      0000G           JSR     IOCLNR
                                ;
                                ; THIS IS THE SECOND PART OF WBY N. PATCH AND SERVES ONLY TO LINK
                                ; TO THE REAL PATCH SPACE
                                ;
                                        GLOBL   WBYP3
0057    7E      0000G   WBYP2:  JMP     WBYP3
```

```
                                        .SBTTL  WBYTVL--OUTPUT ONE BYTE ONTO BUS
                                ;THIS DOES THE ACTUAL TRANSFER AND INTERPRETITAION OF THE INTERGERS TO
                                ;BE TRANSFERED ON THE BUS.
                                ;
                                ;NOTE--- WILL SET EOI ON ARG. ) 256 THRU 512 BYTE WILL BE
                                ;               RESIDUE 256
                                
                                        .GLOBL  WBYTVL
                                        .GLOBL  EOION, EOIOFF, IECSND, ERWBYT
005A    BD      0000G   WBYTVL: JSR     INTSRC          ; SET UP BUS FOR TRANSFER
005D    DE      ^0G             LDX     PCINT,D         ;GET THE F.P. NUMBER
005F    BD      0000G           JSR     PSHFPN
0062    30                      TSX
0063    BD      0^^0G           JSR     FIX1
0066    64      03              LSR     FIX1A,X         ; 0->MSB->C
0068    26      13              BNE     WBVER           ; IF OVER THEN ERROR
006A    25      03              BCS     WBEOI           ; ARG. ) 255. SO SET EOI
006C    4D                      TST A                   ; TEST FOR NEG. ARG.
006D    27      06              BEQ     WBGO
006F    7F      0000G   WBEOI:  CLR     ERRCD
0072    BD      0000G           JSR     EOION           ; TURN ON EOI LINE
0075    E6      04      WBGO:   LDA B   FIX1B,X         ; GET THE NUMBER
0077    BD      0000G           JSR     IECSND
007A    20      05              BRA     WBFINI
                                .BLKB   1               ; THAT'S TO TAKE UP SPACE
007D    B6      006     WBVER:  LDA A   ERWBYT,I
007F    97      006             STA A   ERRCD,D
0081    30              WBFINI: TSX
0082    BD      0000G           JSR     A9X             ;CLEAN UP S.TACK
0085    35                      TXS
0086    39                      RTS
```

```
                                        .SBTTL  RBYTVL--READ ON VALUE OFF OF THE IEC BUS
                                ;THIS ROUTINE READS BYTES OFF OF THE IEC BUS, AND CONVERTS THEM
                                ;TO INTERNAL FPNS.
                                        .GLOBL  RBYTVL, IECRD, FLOAT1, PULFPN
                                        .GLOBL  FPNEG
0087    BD      0000G   RBYTVL: JSR     IECRD           ; GET BYTE
008A    37                      PSH B                   ; DATA BYTE
008B    96      00G             LDA A   CRSTAT.D        ; EOI?
008D    27      05              BEQ     1$
008F    4F                      CLR A
0090    6D                      TST B                   ; 0 AND EOI ?
0091    26      01              BNE     1$
0093    4C                      INC A                   ; MAKE IT 256
0094    36              1$:     PSH A                   ; MS BYTE
0095    36                      PSH A                   ; THAT'S A NICE TAG
0096    BD      0000G           JSR     FLOAT1
0099    96      00G             LDA A   CRSTAT.D        ; EOI?
009B    27      03              BEQ     2$
009D    BD      0000G           JSR     FPNEG           ; NEGATE IT
00A0    DE      00G     2$:     LDX     POINT.D         ; WHERE TO STORE
00A2    BD      0000G           JSR     PULFPN          ; THATS WHERE
00A5    39                      RTS
                0001'           .END
```

```
BYTE- RBYTE, WBYTE I   RT-11 MACRO  VM02-10   14-OCT-76 01:29:17 PAGE 4+
SYMBOL TABLE

ABRFLG=  0040        AFAIL =  0030        AMAT  =  0010        ARRAY =  0020        ASTR  =  0020
ATLOD =  0008        ATMOFF=  xxxxxx G    ATMON =  xxxxxx G    ATSMTG=  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
BERSTI=  0020        BLINK =  xxxxxx G    BMAT  =  0004        BRKCNT=  xxxxxx G    BSTR  =  0008
BUSACT=  0010        CDOPTR=  xxxxxx G    CDSPTR=  xxxxxx G    CHAR  =  xxxxxx G    CHRCNT=  xxxxxx G
CLPTR =  xxxxxx G    CMAT  =  0001        COLCNT=  xxxxxx G    CRDC3 =  0002        CREOF =  0008
CREOI =  0004        CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT=  xxxxxx G
CRVLD =  0080        CSTR  =  0002        CTKN  =  xxxxxx G    CURSOR=  xxxxxx G    DATDEV=  0022
DIMFLG=  0004        DIRCT =  0080        DISCNT=  xxxxxx G    DISSRQ=  0008        DL    =  xxxxxx G
DP    =  xxxxxx G    DREXTA=  xxxxxx G    DREXTB=  xxxxxx G    DSPDEV=  0020        DT    =  xxxxxx G
EDTBFR=  xxxxxx G    ENDKEY=  0040        EOFTYP=  0038        EOIOFF=  xxxxxx G    EOION =  xxxxxx G
EOLTG =  xxxxxx G    EOSTG =  xxxxxx G    ERATSN=  xxxxxx G    ERDOMN=  xxxxxx G    EREOM =  xxxxxx G
FREBER=  xxxxxx G    FREILE=  xxxxxx G    FREIXH=  xxxxxx G    ERIOE =  xxxxxx G    ERNIOD=  xxxxxx G
ERNSEP=  xxxxxx G    ERRCD =  xxxxxx G    ERTERM=  xxxxxx G    ERUNDF=  xxxxxx G    ERWBYT=  xxxxxx G
ESTG  =  xxxxxx G    EXTFLG=  0080        FILDEV=  0000        FIX1  =  xxxxxx G    FIX1A =  0003
FIX1B =  0004        FLOAT1=  xxxxxx G    FMTWLD=  0008        FNFLG =  001C        FORTG =  xxxxxx G
FPNEG =  xxxxxx G    GLBFLG=  xxxxxx G    GUSTG =  xxxxxx G    IECRD =  xxxxxx G    IECSND=  xxxxxx G
IMXTG =  xxxxxx G    INPUTK=  0001        INTACP=  xxxxxx G    INTSRC=  xxxxxx G    IOBFR1=  xxxxxx G
IOCLWP=  xxxxxx G    IOFLGS=  xxxxxx G    I/OFUNC=  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        PGMLNK=  0007        PGMPTR=  xxxxxx G    PGMTG =  xxxxxx G    PIALT =  xxxxxx G
PLOSTG=  xxxxxx G    PNDEOF=  xxxxxx G    PNDFLG=  xxxxxx G    PNTNTG=  xxxxxx G    PNTSTG=  xxxxxx G
POINT =  xxxxxx G    PPMODE=  xxxxxx G    PRIDEF=  0001        PRTTG =  xxxxxx G    PSCTG =  xxxxxx G
PSHFPN=  xxxxxx G    PULFPN=  xxxxxx G    RBYTE    0037RG      RBYTVL   0087RG      RECLFG=  0004
REAIL =  00C0        RMAT  =  0040        RPTCTL=  0080        RSTR  =  0080        RTRNTG=  xxxxxx G
RUNFLG=  0080        RUNN  =  0002        R0    =  xxxxxx G    R1    =  xxxxxx G    R10   =  xxxxxx G
R11   =  xxxxxx G    R12   =  xxxxxx G    R13   =  xxxxxx G    R14   =  xxxxxx G    R15   =  xxxxxx G
R16   =  xxxxxx G    R17   =  xxxxxx G    R18   =  xxxxxx G    R19   =  xxxxxx G    R2    =  xxxxxx G
R20   =  xxxxxx G    R21   =  xxxxxx G    R22   =  xxxxxx G    R23   =  xxxxxx G    R3    =  xxxxxx G
R4    =  xxxxxx G    R5    =  xxxxxx G    R6    =  xxxxxx G    R7    =  xxxxxx G    R8    =  xxxxxx G
R9    =  xxxxxx G    SBP   =  xxxxxx G    SCALER=  0040        SECDEF=  0002        SEMITG=  xxxxxx G
SNOIT =  0080        SPCI08=  xxxxxx G    STAT37=  xxxxxx G    STPFLG=  0040        STPKEY=  0020
STRING=  0010        SYSERR=  xxxxxx G    TABPTR=  xxxxxx G    TABPTR=  xxxxxx G    TCOL  =  xxxxxx G
TRCFLG=  0020        T51DEV=  0025        UNDOR =  xxxxxx G    UNDEF =  0080        VALTG =  xxxxxx G
VALUND=  0040        WBE01    006FR       WBEXT    002FR       WBFINI   0081R       WBGO     0075R
WBNATN   002AR       WBVER    007DR       WBYP1    0032RG      WBYP2    0057R       WBYP3 =  xxxxxx G
WBYTE    0000RG      WBYTVL   005ARG      XAXIS =  xxxxxx G

  ABS.    0000     00
          0096     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 2555  WORDS
.SY:BYTE/C/DK1:SETCLI.BYTE
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03.)

```
ENDKEY  1-213#
EOFTYP  1-186#
EOIOFF  3-9#
EOIGN   3-9#      3-21
EOLTG   1-142#    1-287     2-3
EOSTG   1-143#
ERRTSN  1-170#
ERDOMN  1-164#
EREGM   1-171#
ERFBFR  1-167#
ERFILE  1-169#
ERFIXN  1-282#
ERIOE   1-168#
ERNIOD  1-173#
ERNSEP  1-165#
ERRCD   1-42#     3-20*     3-27*
ERTERM  1-166#
ERUNDF  1-172#
ERWRYT  3-9#      3-26
ESTG    1-124#
EXTFLG  1-25#
FILDEV  1-254#
FIX1    1-282#    3-14
FIX1A   1-117#    3-15*
FIX1B   1-118#    3-22
FLOAT1  4-4#      4-16
FMTVLD  1-203#
ENFLG   1-27#
FORTG   1-126#
FPNEG   4-5#      4-19
GLBFLG  1-71#
GOSTG   1-125#
IECRD   4-4#      4-6
IECSND  3-9#      3-23
IMXTG   1-129#
INPUTK  1-219#
INTACP  1-280#    2-11
INTSRC  1-279#    3-10
IOBFR1  1-266#
IOCLNR  1-280#    1-311     2-18
IOFLGS  1-246#
IOFUNC  1-190#    1-296*    2-13*
ITM1TG  1-136#
ITM2TG  1-137#
ITTDEV  1-260#
JMPX    1-55#
KBDEV   1-255#
KBFLAG  1-210#
KBIN    1-208#
KEYFLG  1-35#
KEYSTK  1-52#
LBRKTG  1-139#
LCIFLG  1-24#
LDAX    1-65#
LDXX    1-60#
LENGTH  1-240#
LISTTG  1-127#
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
LNNOTG  1-135#
LOCTG   1-145#   1-280#   1-303
LSP     1-46#
LSTELI  1-227#
MTBFR   1-267#
MTPD2   1-259#
MTPDEV  1-257#
NLPTR   1-21#
NOKEY   1-209#
NOOUT   1-225#
NOWRIT  1-226#
NTAPTR  1-93#
NTATTR  1-80#
NTDIMS  1-92#
NTOLEN  1-95#
NTLINK  1-27#
NTNAME  1-78#
NTPTR   1-22#
NTRELV  1-215#
NTSPTR  1-97#
NTVAL   1-87#
NTWCOL  1-91#
NTWLEN  1-96#
NTWROW  1-90#
NULLTG  1-122#
OBJATR  1-102#
OBJBCK  1-103#
OBJDT   1-104#
OBJLEN  1-101#
ONSFLG  1-29#
OPRADR  1-238#
PAETG   1-133#
PARM    1-85#
PGMATR  1-109#
PGMBP   1-111#
PGMCD   1-113#
PGMFP   1-110#
PGMLEN  1-108#
PGMLNN  1-112#
PGMPTR  1-48#
PGMTG   1-128#
PIALT   1-279#   1-291    1-293#   2-7      2-9#
PLOSTG  1-123#
PNDEOF  1-45#
PNDFLG  1-44#
PNTNTG  1-132#
PNTSTG  1-130#
POINT   1-241#   3-11     4-20
PPMODE  1-224#   1-304
PRTDEF  1-206#
PRTTG   1-141#
PSCTG   1-131#
PSHFPN  1-282#   3-12
PULFPN  4-4#     4-21
R0      1-38#    1-289#   1-309#   2-5#
R1      1-38#
R10     1-38#
```

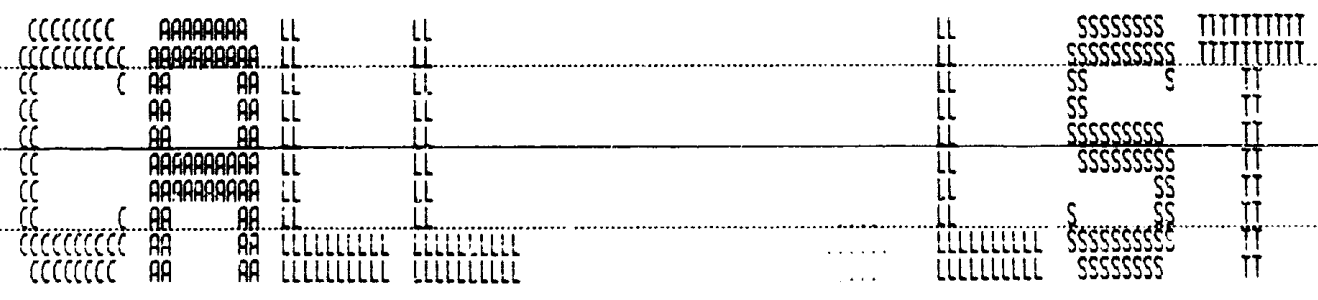
CROSS REFERENCE TABLE (CREF V01-03)

```
R11     1-38#
R12     1-38#
R13     1-38#
R14     1-38#
R15     1-38#
R16     1-38#
R17     1-39#
R18     1-39#
R19     1-39#
R2      1-38#
R20     1-39#    1-305*   1-307
R21     1-39#
R22     1-39#
R23     1-39#
R3      1-38#
R4      1-38#
R5      1-38#
R6      1-38#
R7      1-38#
R8      1-38#
R9      1-38#
RBYTE   1-281*   2-3#
RBYTVL  4-4#     4-6#
RECLFG  1-217#
REGIL   1-152#
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNTG  1-140#
RUNFLG  1-32#
RUNN    1-218#
SBP     1-47#
SCALER  1-82#
SECDEF  1-205#
SEMITG  1-144#
SNDIT   1-228#
SPCIOB  1-280#   1-306    1-310    2-17
STAT37  1-247#
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSERR  1-43#
T51DEV  1-261#
TABPTR  1-243#
TAGPTR  1-239#
TCOL    1-245#
TRCFLG  1-34#
UNADR   1-279#
UNDEF   1-81#
VALTG   1-134#
VALUND  1-88#
WBEOI   3-17     3-20#
WBEXT   1-308    1-311#
WBFIN!  3-24     3-28#
WBGO    3-19     3-22#
WBNATN  1-299    1-309#
WBVER   3-16     3-26#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-36



```
                                        .TITLE  CALLBS
                                        .IDENT  /JMBE08/
                                ;
                                ;       CALL CHANGES THE STRING STACK ENTRY THAT MUST BE ON BOTTOM
                                ;       OF STACK TO A CALL STACK ENTRY
                                ;
                                ;       CALL STACK ENTRY FORMAT
                                ;       LENGTH  USE
                                ;       1       ENTRY TAG (CALLTG)
                                ;       1       BANK CALL WAS CALLED FROM
                                ;       1       MPU CONDITION CODE REGISTER WHEN CALL WAS CALLED
                                ;       2       RETURN POINT FOR CALLMODULE TO RETURN TO
                                ;
                                ;       BANK SWITCH CONTROL REGISTER F   ;  ;(=DON'T CARE)
                                ;       XX 000 XXX - MAIN ROM
                                ;       XX 001 XXX - EXTENDED FUNCTIO
                                ;       XX 010 XXX - COMM.PACK ROM
                                ;       XX 011 XXX - NOT ASSIGNED
                                ;       XX 100 XXX - LEFT PACK
                                ;       XX 101 XXX - RIGHT PACK
                                ;       XX 110 ZZZ - LEFT TOASTER
                                ;       XX 111 ZZZ - RIGHT TOASTER
                                ;
                                ;       ZZZ = TOASTER SELECTION BITS
                                ;
                                        .GLOBL  CALL,CALLBS
                0000'           CALLBS  =       .
                                        .GLOBL  SETBNK,BNKADR,ERRCD,ERCNSF,ITM1TG,A8X
                                        .GLOBL  LDAX,STAX,UPCASE,CALLTG
                                        .GLOBL  RTRNTG,EOLTG,CLRARG,D5X,LOCTG,BANK
                                        .GLOBL  R0,R1,R2,R3,R4,R5,R6
0000    30              CALL:   TSX
0001    08                      INX
0002    DF      00G             STX     R0,0
0004    86      00G             LDA A   EOLTG,1         ;SEARCH FOR END OF LINE
0006    BD      0000G           JSR     LOCTG
0009    BD      0000G           JSR     D5X             ;POINTER TO STRING NT OR LETREL
000C    DF      00G             STX     R5,0
000E    DF      00G             STX     R0,0
0010    BD      0000G           JSR     CLRARG          ;GET TYPARG TO CHECK ON IT
0013    81      20              CMP A   32.,1
0015    26      34              BNE     00PS            ;FORCE TO BE STRING, ELSE ERROR
0017    DE      00G             LDX     R5,0
0019    EE      04              LDX     4,X             ;POINTER TO LENGTH
001B    A6      00              LDA A   0,X
001D    26      06              BNE     1$              ;>6
001F    E6      01              LDA B   1,X
0021    C1      07              CMP B   7,1
0023    25      02              BCS     2$              ;<=6
0025    C6      06      1$:     LDA B   6,1             ;ASSUME 6
0027    08              2$:     INX                     ;OFF BY ONE, BUT SO WHAT
                                ;THE END JUSTIFIES THE MEANS???
0028    DF      00G             STX     R6,0
002A    CE      2020            LDX     1+256.*32.,1    ; A FEW SPACES
002D    DF      00G             STX     R1,0
002F    DF      00G             STX     R2,0
0031    DF      00G             STX     R3,0
```

```
                                
    73  0033  17          3$:   TBA
    74  0034  27  1A            BEQ   BE
    75  0036  DE  00G           LDX   R6,0          ;POINTER TO SOURCE, MORE OR LESS
    76  0038  BD  0C00G         JSR   LDAX
    77  003B  BD  0000G         JSR   UPCASE
    78  003E  36                PSH A
    79  003F  17                TBA
    80  0040  33                PUL B               ;SWAP FOR SOME REASON
    81  0041  CE  FFFFG         LDX   R1-1,I        ;DEXTINATION FEILD
    82  0044  BD  0000G         JSR   STAX
    83  0047  1C                TAB                 ;RESTORE B BACK TO WHAT IT WAS
    84  0048  5A                DEC B
    85  0049  20  E8            BRA   3$
    86                          ;
    87  004B  86  00G   OOPS:   LDA A  ERCNSF,I
    88  004D  97  00G           STA A  ERRCD,0
    89  004F  39                RTS
    90                          ;
    91  0050  DE  00G   BE:     LDX   R5,0          ;CHANGE STRING TO CALL STACK ENTRY
    92  0052  32                PUL A
    93  0053  A7  04            STA A  4,X          ;WHERE I RETURN
    94  0055  32                PUL A
    95  0056  A7  05            STA A  5,X
    96  0058  07                TPA
    97  0059  A7  03            STA A  3,X          ;SAVE MPU STATUS
    98  005B  96  00G           LDA A  BANK,0
    99  005D  A7  02            STA A  2,X          ;CURRENT BANK
   100  005F  86  00G           LDA A  CALLTG,I
   101  0061  A7  01            STA A  1,X          ;NOW IT IS CALL STACK ENTRY
   102  0063  4F                CLR A
   103  0064  97  00G   LOOPA:  STA A  R4,0          ;SAVE BANK ADDR
   104  0066  BD  0000G         JSR   SETBNK
   105  0069  FE  0000G         LDX   BNKADR        ;TEST DATA ID
   106  006C  8C  4051          CPX   16465.,I
   107  006F  26  36            BNE   SKIPA
   108  0071  CE  0000G         LDX   BNKADR+13.,I  ;FIRST NAME ADDR
   109  0074  DF  00G   LOOPB:  STX   R5,0          ;MS ADDR
   110  0076  A6  00            LDA A  0,X          ;IS THIS EOD
   111  0078  27  2D            BEQ   SKIPA         ;END OF THIS BANK
   112  007A  EE  00            LDX   0,X           ;TEST TWO BYTES AT A TIME
   113  007C  9C  00G           CPX   R1,0
   114  007E  26  3C            BNE   SKIPB
   115  0080  DE  00G           LDX   R5,0
   116  0082  EE  02            LDX   2,X
   117  0084  9C  00G           CPX   R2,0
   118  0086  26  34            BNE   SKIPB
   119  0088  DE  00G           LDX   R5,0
]]]]]
   120  008A  EE  04            LDX   4,X
   121  008C  9C  00G           CPX   R3,0
   122  008E  26  2C            BNE   SKIPB
   123  0090  DE  00G           LDX   R5,0          ;I GOT A HIT
   124  0092  EE  06            LDX   6,X
   125  0094  AD  00            JSR   0,X
   126  0096  9F  00G   EXIT:   STS   R0,0          ;LOCATE CALL ENTRY
   127  0098  86  00G           LDA A  CALLTG,I
   128  009A  BD  0000G         JSR   LOCTG
   129  009D  9E  00G           LDS   R0,0
```

```
 130  009F  32                        PUL A                   ;CALLTG
 131  00A0  32                        PUL A                   ;CALLERS BANK ADDRESS
 132  00A1  BD  0000G                 JSR     SETBNK
 133  00A4  32                        PUL A                   ;OLD CC REG
 134  00A5  06                        TAP
 135  00A6  39                        RTS                     ;SEE YOU LATER
 136                          ;
 137  00A7  96  00G       SKIPA:      LDA A   R4,D            ;GET TEST BANK ADDR
 138  00A9  81  30                    CMP A   48.,I           ;IF TOASTER SECTION INC BY ONE
 139  00AB  25  08                    BCS     SETNXT          ;OF BANK ZERO TRY AGAIN
 140  00AD  4C                        INC A
 141  00AE  81  40                    CMP A   64.,I           ;TIME TO STOP
 142  00B0  26  B2                    BNE     LOOPA
 143  00B2  86  00G                   LDA A   ERCNSF,I
 144  00B4  97  00G                   STA A   ERRCD,D
 145  00B6  20  0E                    BRA     EXIT
 146                          ;
 147  00B8  8B  08        SETNXT:     ADD A   8.,I            ;NEXT BANK ADDR AFTER BANK 0
 148  00BA  20  A8                    BRA     LOOPA
 149                          ;
 150  00BC  DE  00G       SKIPB:      LDX     R5,D            ;BUMP MS ADDR
 151  00BE  BD  0000G                 JSR     A8X
 152  00C1  20  B1                    BRA     LOOPB
 153                          ;
 154       0001'                      .END
```

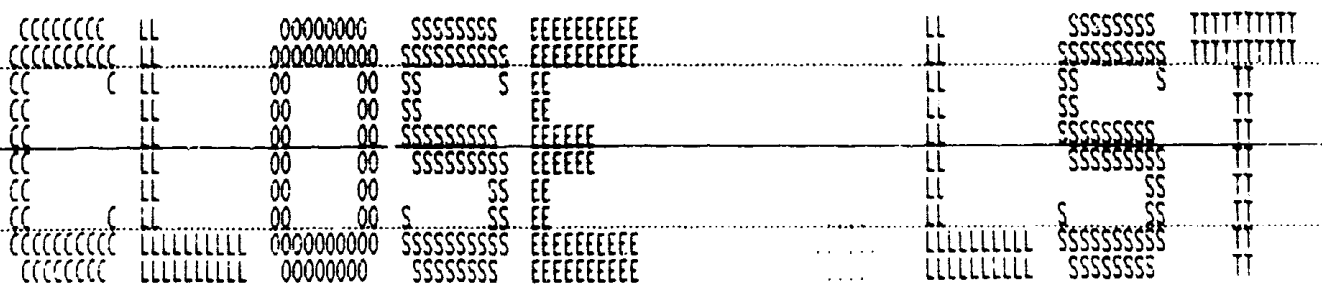
```
CALLBS RT-11 MMAC  VM02-10   14-OCT-76 01:29:27 PAGE 1+
.SYMBOL TABLE

A8X    = XXXXXX G      BANK   = XXXXXX G      BE       0050R      BNKADR= XXXXXX G      CALL     000CRG
CALBS=   0000RG        CALLTG= XXXXXX G       CLRARG= XXXXXX G      DSX    = XXXXXX G      EOLTG = XXXXXX G
ERCNSF= XXXXXX G       ERRCD  = XXXXXX G      EXIT     0096R      ITM1TG= XXXXXX G      LOAX   = XXXXXX G
LOCTG = XXXXXX G       LOOPA    0064R         LOOPB    0074R      OOPS     004BR        RTRNTG= XXXXXX G
R0     = XXXXXX G      R1     = XXXXXX G      R2     = XXXXXX G      R3     = XXXXXX G      R4     = XXXXXX G
R5     = XXXXXX G      R6     = XXXXXX G      SETBNK= XXXXXX G       SETNXT   00B8R        SKIPA    00A7R
SKIPB    00BCR         STAX   = XXXXXX G      UPCASE= XXXXXX G
  ABS.   0000    00
         00C3    01
ERRORS DETECTED: 0      WARNINGS POSTED: 0      FREE CORE: 3353. WORDS
.SY:CALL/C(DK1:SETCLI.CALL
```

CROSS REFERENCE TABLE (CREF V01-03.)

	1-42								
A8X	1-43#	1-151							
BANK	1-45#	1-98							
BE	1-74	1-31#							
BNKADR	1-43#	1-105	1-108						
CALL	1-41#	1-47#							
CALLBS	1-41#	1-42#							
CALLTG	1-44#	1-100	1-127						
CLRP2G	1-45#	1-55							
D8X	1-45#	1-52							
EOLTG	1-45#	1-50							
ERCNSF	1-43#	1-87	1-143						
ERRCD	1-43#	1-88%	1-144%						
EXIT	1-126#	1-145							
ITM1TG	1-43#								
LDAX	1-44#	1-76							
LOCTG	1-45#	1-51	1-128						
LOOPA	1-103#	1-142	1-148						
LOOPB	1-109#	1-152							
OOPS	1-57	1-87#							
R0	1-46#	1-49%	1-54%	1-126%	1-129				
R1	1-46#	1-70%	1-81	1-113					
R2	1-46#	1-71%	1-117						
R3	1-46#	1-72%	1-121						
R4	1-46#	1-103%	1-137						
R5	1-46#	1-53%	1-58	1-91	1-109%	1-115	1-119	1-123	1-150
R6	1-46#	1-68%	1-75						
RTRNTG	1-45#								
SETBNK	1-43#	1-104	1-132						
SETNXT	1-139	1-147#							
SKIPA	1-107	1-111	1-137#						
SKIPB	1-114	1-118	1-122	1-150#					
STAX	1-44#	1-82							
UPCASE	1-44%	1-77							

SE1 1-7#


CCCCCCCC LL 00000000 SSSSSSSS EEEEEEEEEE LL SSSSSSSS TTTTTTTTTT
CCCCCCCCCC LL 0000000000 SSSSSSSSSS EEEEEEEEEE LL SSSSSSSSSS TTTTTTTTTT
CC C LL 00 00 SS S EE LL SS S TT
CC LL 00 00 SS EE LL SS TT
CC LL 00 00 SSSSSSSS EEEEEE LL SSSSSSSS TT
CC LL 00 00 SSSSSSSS EEEEEE LL SSSSSSSS TT
CC LL 00 00 SS EE LL SS TT
CC C LL 00 00 S SS EE LL S SS TT
CCCCCCCCCC LLLLLLLLLL 0000000000 SSSSSSSSSS EEEEEEEEEE LLLLLLLLLL SSSSSSSSSS TT
CCCCCCCC LLLLLLLLLL 00000000 SSSSSSSS EEEEEEEEEE LLLLLLLLLL SSSSSSSS TT

CLOSE RT-11 MMAC VM02-10 14-OCT-76 01:29:33
TABLE OF CONTENTS

```
                                        .TITLE  CLOSE
                                        .IDENT  /SBR006/
                                        .GLOBL  MTCLOS
                                        .GLOBL  INITMT,CRTRST,CLOSE
                                        .GLOBL  CTKN            ; CURRENT TOKEN
                                        .GLOBL  OLDCOD          ; "OLD"
                                        .GLOBL  BNKADR,RUNBNK,HALTR
                                ;
                                        .GLOBL  SETERR,ERATSN,VALTG,CLOSP1
0000    01              CLOSE:  NOP                     ; THIS MAKES THE PATCH FIT
0001    CE      0008G           LDX     BNKADR+8.,I     ; RUN THE BANKS
0004    BD      0000G           JSR     RUNBNK
0007    96      00G             LDA A   CTKN,D          ; OLD DOESN'T CLOSE MAGTAPE
0009    81      00G             CMP A   OLDCOD,I
000B    27      23              BEQ     COPRTS
000D    30                      TSX                     ; VALUE TAG
000E    A6      02              LDA A   2,X
0010    81      00G             CMP A   VALTG,I
0012    27      06              BEQ     CLOERR
0014    6D      0000G           JSR     INITMT          ; CLOSE MAGTAPE
0017    7E      0000G           JMP     CLOSP1          ; THIS HERE IS A CALL TO THE PATCH!!
001A    BD      0000G   CLOERR: JSR     SETERR
001D    00G                     .BYTE   ERATSN
                                        .SBTTL  COPY----THIS IS THE COPY COMMAND
                                        .GLOBL  COPY,DSPCPY,FILES,COPTYP

                                ;THIS IS THE COPY COMMAND.    IF IT IS A FILE FORM OF THE COMMAND,
                                ;THE FILE PROCESSOR IS BRANCHED TO.

001E    CE      0000G   COPY:   LDX     COPTYP,I
0021    DF      00G             STX     OPRADR,D
0023                            TSX
0024            02              LDA A   2,X             ;LOOKFOR EOL TAG. IF THERE
0026    81      00G             CMP A   EOLTG,I         ;THERE?
0028    27      03              BEQ     1$              ;YES, COPY THE SCREEN ONLY
002A    7E      0000G           JMP     FILES           ; FILE SYSTEM CALL
002D    BD      0000G   1$:     JSR     DSPCPY          ;COPY SCREEN
0030    39              COPRTS: RTS

                0001'                   END
```

```
CLOSE  RT-11 MMAC  VM02-10   14-OCT-76 01:29:33 PAGE 2+
SYMBOL TABLE

ABPFLG=  0040        AFAIL =  0030        AMAT  =  0010        ARRAY =  0020        ASTR  =  0020
ATLOD =  0008        ATSWTG=  XXXXXX G    ATWLD =  0004        A.END =  XXXXXX G    A.MRK =  XXXXXX G
A.PRIM=  XXXXXX G    A.PTR =  XXXXXX G    A.SEC =  XXXXXX G    A.STAT=  XXXXXX G    A.STRT=  XXXXXX G
BAKSTG=  XXXXXX G    BANK  =  XXXXXX G    BFRSTT=  0020        BLINK =  XXXXXX G    BMAT  =  0004
BNKADR=  XXXXXX G    BRKCNT=  XXXXXX G    BSTR  =  0008        BUSACT=  0010        CDOPTR=  XXXXXX G
CDSPTR=  XXXXXX G    CHGR  =  XXXXXX G    CHRCNT=  XXXXXX G    CLOERR   001AR       CLOSE    0000RG
CLOSP1=  XXXXXX G    CLPTR =  XXXXXX G    CMAT  =  0001        COLCNT=  XXXXXX G    COPRTS   0030R
COPTYP=  XXXXXX G    COPY     001ERG      CRDC3 =  0002        CREOF =  0008        CREOI =  0004
CREOT =  0020        CRETX =  0010        CRNORM=  0001        CRSTAT=  XXXXXX G    CRTRST=  XXXXXX G
CRVLD =  0080        CSTR  =  0002        CTKN  =  XXXXXX G    CURSOR=  XXXXXX G    DATDEV=  0022
DIMFLG=  0004        DIRCT =  0080        DISCNT=  XXXXXX G    DISSRC=  0008        DL    =  XXXXXX G
DP    =  XXXXXX G    DREXTA=  XXXXXX G    DREXTB=  XXXXXX G    DSPCPY=  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    ERFBER=  XXXXXX G
ERFILE=  XXXXXX G    ERIOE =  XXXXXX G    ERNIOC=  XXXXXX G    ERNSEP=  XXXXXX G    ERRCD =  XXXXXX G
ERTERM=  XXXXXX G    ERUNDF=  XXXXXX G    ESTG  =  XXXXXX G    EXTFLG=  0080        FILDEV=  0000
FILES =  XXXXXX G    FIX1A =  0003        FIX1B =  0004        FMTWLD=  0008        FNELG =  0010
FORTG =  XXXXXX G    GLBFLG=  XXXXXX G    GOSTG =  XXXXXX G    HALTR =  XXXXXX G    IMXTG =  XXXXXX G
INITMT=  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
MTCLOS=  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    NTRELV=  0010
NTSPTR=  0008        NTVAL =  0005        NTWCOL=  0007        NTWLEN=  0007        NTWROW=  0005
NULLTG=  XXXXXX G    OBJATR=  0002        OBJBCK=  0003        OBJOT =  0005        OBJLEN=  0000
OLDCOD=  XXXXXX G    ONSFLG=  0002        OPRADR=  XXXXXX G    PDETG =  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
PNOFLG=  XXXXXX G    PNTNTG=  XXXXXX G    PNTSTG=  XXXXXX G    POINT =  XXXXXX G    PPMODE=  XXXXXX G
PRIDEF=  0001        PRTTG =  XXXXXX G    PSCTG =  XXXXXX G    RECLFG=  0004        RFAIL =  00C0
RMAT  =  0040        RPTCTL=  0080        RSTR  =  0080        RTRNTG=  XXXXXX G    RUNBNK=  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
SETERR=  XXXXXX G    SNDIT =  0080        STAT32=  XXXXXX G    STPFLG=  0040        STPKEY=  0020
STRING=  0010        SYSERR=  XXXXXX G    TABPTR=  XXXXXX G    TAGPTR=  XXXXXX G    TCOL  =  XXXXXX G
TRCFLG=  0020        T51DEV=  0025        UNDEF =  0080        VALTG =  XXXXXX G    VALUNO=  0040
XAXIS =  XXXXXX G

    ABS.    0000    00
            0031    01

ERRORS DETECTED: 0      WARNINGS POSTED: 0      FREE CORE: 2627  WORDS
.SY:CLOSE/C<DK1:SETCLI,CLOSE
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

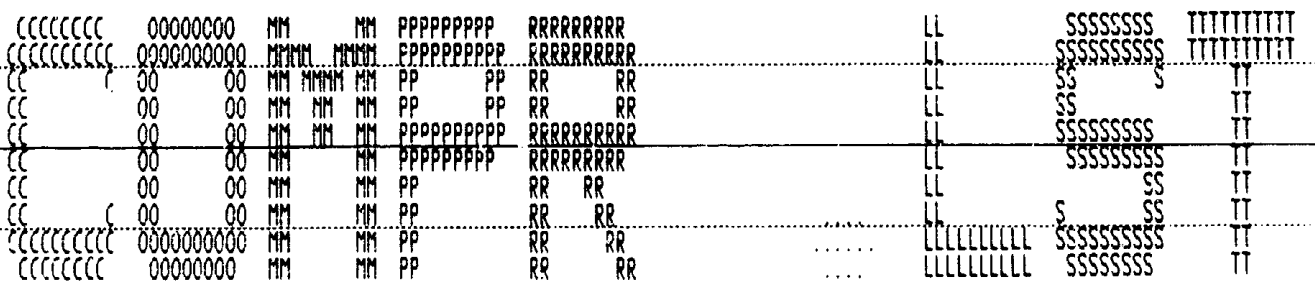
LSTEMT	1-227#	
MTBFR	1-267#	
MTCLOS	2-3#	
MTPD2	1-259#	
MTPDEV	1-257#	
NLPTR	1-21#	
NOKEY	1-209#	
NOOUT	1-225#	
NOWRIT	1-226#	
NTAPTR	1-93#	
NTATTR	1-80#	
NTDIMS	1-92#	
NTDLEN	1-95#	
NTLINK	1-77#	
NTNAME	1-78#	
NTPTR	1-22#	
NTRELY	1-215#	
NTSPTR	1-97#	
NTVOL	1-82#	
NTWCOL	1-91#	
NTWLEN	1-96#	
NTWROW	1-90#	
NULLTG	1-122#	
OBJATR	1-102#	
OBJBCK	1-103#	
OBJDT	1-104#	
OBJLEN	1-101#	
OLDCOD	2-6#	2-14
OMSFLG	1-29#	
OPRADR	1-238#	2-31#
PARTG	1-133#	
PARM	1-85#	
PGMATR	1-109#	
PGMBP	1-111#	
PGMCD	1-113#	
PGMFP	1-110#	
PGMLEN	1-108#	
PGMLNN	1-112#	
PGMPTR	1-48#	
PGMTG	1-128#	
PLOSTG	1-123#	
PMDEOF	1-45#	
PNDFLG	1-44#	
PNTNTG	1-132#	
PNTSTG	1-130#	
POINT	1-241#	
PPMODE	1-224#	
PRIDEF	1-206#	
PRTTG	1-141#	
PSCTG	1-131#	
R0	1-38#	
R1	1-38#	
R10	1-38#	
R11	1-38#	
R12	1-38#	
R13	1-38#	
R14	1-38#	

CROSS REFERENCE TABLE (CREF V01-03)

```
R15     1-38#
R16     1-38#
R17     1-39#
R18     1-39#
R19     1-39#
R2      1-38#
R20     1-39#
R21     1-39#
R22     1-39#
R23     1-39#
R3      1-38#
R4      1-38#
R5      1-38#
R6      1-38#
R7      1-38#
R8      1-38#
R9      1-38#
RECLFG  1-217#
REQIL   1-152#
RMAT    1-151#
RPTCTL  1-212#
RSTR    1-150#
RTRNTG  1-140#
RUNBNK  2-7#      2-12
RUNFLG  1-32#
RUNN    1-218#
SBP     1-47#
SCALER  1-82#
SECDEF  1-205#
SEMITG  1-144#
SETERR  2-9#      2-22
SNDIT   1-228#
STAT37  1-247#
STPFLG  1-33#
STPKEY  1-214#
STRING  1-84#
SYSERR  1-43#
T51DEV  1-261#
TABPTR  1-243#
TAGPTR  1-239#
TCOL    1-245#
TRCFLG  1-34#
UNDEF   1-81#
VALTG   1-134#    2-9#      2-18
VALUND  1-88#
XAXIS   1-248#
```

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3#



14-OCT-76

```
                                .IDENT  /MCD809/
                                .GLOBL  COMPR
0000'                           COMPR=.
                                .TITLE  COMPR
                                .GLOBL  BUG16A,BUG16B,BUG16C,USRORG,ZX
                                .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7
                                .GLOBL  LSP,ERWSFL,ERRCD,CLPTR,BACKUP
                                .GLOBL  LOCTG,PGMTG,STPTR,PGMPTR,XBFLAG,RTRNTG
                                .GLOBL  PLOSTG,ESTG,GOSTG,FORTG,LISTTG,IMXTG
                                .GLOBL  PRTTG,LOCTGR,PAETG,DISBLE,ENABLE,SAFE
                                .GLOBL  CLOPTR,COSPTR,NTPTR,NLPTR


                        ;       COMPR COLLECTS AT THE LOWEST POSSIBLE ADDRESSES ALL THE
                        ;       VALID ARRAYS, STRINGS, AND PROGRAM LINES BETWEEN PGMEND AND
                        ;       LSP AND MOVES LSP DOWN AS FAR AS POSSIBLE.  IN THAT WAS IT
                        ;       COLLECTS ABOVE LSP IN THE FREE MEMORY SPACE, AS MUCH
                        ;       SPACE AS POSSIBLE.
                        ;
                        ;       INPUT:  THE MINIMUM NUMBER OF BYTES NEEDED TO BE ADDED TO
                        ;               TO THE FREE SPACE AREA FOR THE COMPRESS TO BE CONSIDERED
                        ;               SUCCESSFUL.
                        ;
                        ;       OUTPUTS: NONE.
                        ;
                        ;       ERRORS: ERWSFL  -WORK SPACE FULL, IF THE COMPRESS FAILS TO
                        ;                       RECOVER THE REQUESTED MINIMUM OR MORE.
                        ;
                        ; REGISTER USAGE
                        ; R0 USED BY LOCTG AND LOCTGR
                        ; R4 USED BY LOCTG AND LOCTGR
                        ; R4 POINTER TO ITEM BEING EXAMINED FOR VALIDITY AND POSSIBLY MOVED
                        ; R3 POINTER TO WHERE TO PUT THIS VALID ITEM (ESTABLISHED WHEN ONE IS
                        ;    FOUND AND R2 IS MOVED)
                        ; R2 POINTER TO WHERE TO PUT THE NEXT VALID ITEM
                        ; R5 USED DURING BLOCK MOVE TO POINT
                        ;       ; TO WHERE NEXT BYTE GOES
                        ; R6 USED DURING BLOCK MOVE TO POINT
                        ;       ; TO THE NEXT BYTE TO BE MOVED
                        ;
0000    86      000G            LDA A   RTRNTG,I
0002    36                      PSH A                   ; TAG THE RET. ADDR.
0003    BD      0000G           JSR     DISBLE          ; SET NOT REALLY
0006    DE      000G            LDX     STPTR,D         ; LOAD NAME TABLE PTR
0008    27      20      VLOOP:  BEQ     PSTRT           ; IF NULL, WE'R DONE WITH THIS STEP
000A    A6      04              LDA A   4,X             ; LOAD NT ATTR BYTE
000C    2B      18              BMI     NEXT            ; IF UNDEF GO TO NEXT NT ENTRY
000E    85      30              BIT A   #30,I ; UNLESS EITHER ARRAY OR STR GO
0010    27      14              BEQ     NEXT            ; TO NEXT NT ENTRY
0012    DF      000G            STX     R0,D            ; SAVE PTR INTO NT
0014    EE      08              LDX     11,X            ; LOAD PTR TO STORAGE
0016    A6      02              LDA A   2,X             ; LOAD ATTR BYTE IN STORAGE
0018    8A      80              ORA A   #80,I ; SET THE VALID BIT
001A    A7      02              STA A   2,X
001C    96      000G            LDA A   R0,D            ; RELOAD PTR INTO NT
```

```
                                        STA A   3,X             ; AND STORE INTO BACK PTR IN
                                        LDA A   R0+1,D          ; DATA ELEMENT
                                        STA A   4,X
                                        LDX     R0,D
                        NEXT:           LDX     0,X
                                        BRA     VLOOP
                                ; PROGRAM LINE VALIDATOR
                        PSTRT:          SIS     R0,G
                                        LDX     CLPTR,D         ; VALIDATE THE CURRENT LINE
                                        BEQ     PLSTRT
                                        LDA A   2,X             ; (IN CASE IT IS AN IMMEDIATE
                                        ORA A   #80,I   ; LINE)
                                        STA A   2,X
                        PLSTRT:         LDX     PGMPTR,D        ;
                        PLOOP:          BEQ     NXTPGM          ; IF NO NEXT LINE LOOK FOR A
                                                                ; "HIDDEN PROGRAM"
                                        LDA A   2,X             ; SET VALID BIT IN THIS LINE
                                        ORA A   #80,I
                                        STA A   2,X
                                        LDX     3,X             ; GET PTR TO NEXT LINE
                                        BRA     PLOOP
                        NXTPGM:         LDA A   PGMTG,I         ; SEARCH FOR A "HIDDEN PROGRAM"
                                        JSR     LOCTG           ; X AND R0 LEFT PTING TO TAG-1 IF FOUND
                                        BEQ     START           ; IF NONE THEN START WORKING
                                        STX     R2,D
                                        JSR     BACKUP
                                        LDX     R2,D
                                        LDX     2,X             ; IF FOUND THEN VALIDATE ITS LINES
                                        BRA     PLOOP
                                ; FIND INVALID ITEMS
                        START:          LDX     USRORG,D        ; SET R2 (LOW PTR) TO THE HI END
                                        NOP
                                        STX     R4,D            ; OF THE SPACE BEING COMPRESSED
                                        STX     R2,D            ; LOAD ADDR OF ONE BEYOND THE
                                                                ; HIGH END OF SYSTEM SPACE
                                        CPX     LSP,D
                                        BNE     IN
                                        JMP     DONE
                        SEARCH:         LDA A   2,X             ; TEST THE VALID BIT IN THIS ITEM
                                        BPL     SKIPNG
                                        AND A   #7F,I   ; IF VALID, RESET VALID BIT
                                        JSR     ADDLEN          ; ADD LENGTH TO X AND LOOK AT
                                        BRA     SEARCH          ; NEXT ITEM
                        SKIPNG:         STX     R4,D            ; SAVE PTR TO HI END OF VALID STUFF
                        GARBGE:         JSR     ADDLEN          ; SKIP TO NEXT ITEM
                        ITEMS:          CPX     LSP,D           ; ARE WE AT THE END NOW
                                        BNE     IN
                                        JMP     DONE            ; IF SO GET OUT
                        IN:             LDA B   2,X             ; IF NOT, CHECK VALID BIT
                                        BPL     GARBGE          ; IF INVALID GO ON
                                        AND B   #7F,    ; IF VALID, RESET VALID BIT THEN GET
                                                                ; READY TO COPY
                                        STA B   2,X             ; THIS ITEM DOWN TO JUST ABOVE THE
                                                                ; VALID STUFF
                                                                ;
                                        LDX     R2,D            ; STORE THE POINTER TO
```

```
0085  DF  00G              STX    R3,D        ; THE SPACE WHERE WE'RE GOING TO PUT
0087  DE  00G              LDX    R4,D        ; THE ITEM IN R3
0089  A6  01               LDA A  1,X         ; SET R2 TO POINT TO THE HI END.
008B  9B  01G              ADD A  R2+1,D      ; OF WHERE WE WANT TO MOVE THE
008D  97  01G              STA A  R2+1,D      ; VALID ITEM (THIS WILL BE THE
008F  A6  00               LDA A  0,X         ; NEW PTR TO THE HI END OF THE
0091  99  00G              ADC A  R2,D        ; VALID STUFF)
0093  97  00G              STA A  R2,D
0095  9C  00G              CPX    R3,D        ; *****---------------*******
0097  26  03               BNE    1$          ; *****---------------*******
0099  7E  0139'            JMP    NONEED      ;
                                              ;    R3      R2      R4
009C  9F  00G       1$:    STS    R0,D        ; SAVE STACK PTR
009E  DF  00G              STX    R6,D
00A0  DE  00G              LDX    R3,D
00A2  DF  00G       KWIK:  STX    R5,D
00A4  DE  00G              LDX    R6,D
00A6  A6  00               LDA A  0,X
00A8  08                   INX
00A9  DF  00G              STX    R6,D
00AB  DE  00G              LDX    R5,D
00AD  A7  00               STA A  0,X
00AF  08                   INX
00B0  9C  00G              CPX    R2,D
00B2  26  EE               BNE    KWIK
00B4  C5  08               BIT B  #08,I       ; TEST IF THIS IS A PROGRAM LINE
00B6  26  03               BNE    3$
00B8  7E  0170'            JMP    NMTPTR      ; IF NO UPDATE NT PTR
00BB  DE  00G       3$:    LDX    R3,D        ; UPDATE BACKPTR OF NEXT LINE
00BD  EE  03               LDX    3,X
00BF  27  08               BEQ    1$
00C1  96  00G              LDA A  R3,D
00C3  A7  05               STA A  5,X
00C5  96  01G              LDA A  R3+1,D
00C7  A7  06               STA A  6,X
00C9  DE  00G       1$:    LDX    R3,D        ; UPDATE FORWARD PTR OF PREVIOUS
00CB  EE  05               LDX    5,X         ; LINE
00CD  26  09               BNE    2$
00CF  DE  00G              LDX    PGMPTR,D
00D1  9C  00G              CPX    R4,D
00D3  26  08               BNE    GOAWN
00D5  CE  FFFDG            LDX    PGMPTR-3,I
00D8  96  00G       2$:    LDA A  R3,D
00DA  A7  03               STA A  3,X
00DC  96  01G              LDA A  R3+1,D
00DE  A7  04               STA A  4,X
00E0  96  00G       GOAWN: LDA A  R3,D
00E2  D6  01G              LDA B  R3+1,D
00E4  DE  00G              LDX    R4,D
00E6  BC  0000G            CPX    CDSPTR
00E9  26  0D               BNE    3$
00EB  B7  0000G            STA A  CDSPTR
00EE  F7  0001G            STA B  CDSPTR+1
00F1  CE  FFFCG            LDX    CDOPTR-4,I
00F4  8D  4E               BSR    PTR2
00F6  20  E8               BRA    GOAWN       ; FIG'UR OUT WHAT THAT DOES, IF
                                              ; YOU CAN
```

```
                                
  187  00F8  9C  00G      3$:   CPX     CLPTR,D
  188  00FA  26  09            BNE     2$
  189  00FC  97  00G           STA A   CLPTR,D
  190  00FE  D7  01G           STA B   CLPTR+1,D
  191  0100  CE  FFEC'         LDX     NTPTR-4,I
  192  0103  8D  3F            BSR     PTR2
  193  0105  9C  00G      2$:   CPX     NLPTR,D
  194  0107  26  04            BNE     LOOP
  195  0109  97  00G           STA A   NLPTR,D
  196  010B  D7  01G           STA B   NLPTR+1,D
  197  010D  86  00G      LOOP: LDA A   PLOSTG,I  ; SET THE TOP AND BOTTOM OF THE
  198  010F  C6  00G           LDA B   IMXTG,I   ; RANGE OF TAGS WE NEED TO FIND
  199  0111  BD  0000G         JSR     LOCTGR    ; AND SCAN THE STACK
  200  0114  9C  00G           CPX     ZX,D      ; IF X=0 WE FOUND NONE OF
  201  0116  27  25            BEQ     RESTRT    ; THESE ITEMS
  202  0118  A6  01            LDA A   1,X       ; IF X<>0 SAVE THE TAG FOUND
  203  011A  EE  02            LDX     2,X       ; AND LOAD THE POINTER FOLLOWING IT
  204  011C  9C  00G           CPX     R4,D      ; IF THIS POINTER=PTR TO THE ITEM
  205  011E  26  37            BNE     EVLCHK    ; WE JUST MOVED CORRECT IT ELSE GO TO
  206                                            ; EVLCHK
  207  0120  DE  00G           LDX     R0,D
  208  0122  D6  00G           LDA B   R3,D
  209  0124  E7  02            STA B   2,X
  210  0126  D6  01G           LDA B   R3+1,D
  211  0128  E7  03            STA B   3,X
  212  012A  81  00G           CMP A   ESTG,I    ; IS THE TAG FOUND AN EVALUATOR
  213  012C  27  08            BEQ     1$        ; STATUS. IF SO GO TO PTR2
  214  012E  81  00G           CMP A   PLOSTG,I  ; IS IT A   TO LITERAL IN OBJ STR, IF
  215  0130  26  25            BNE     EVLCHK    ; NOT GO TO EVLCHK
  216  0132  8D  10            BSR     PTR2      ; ELSE GO TO PTR2
  217  0134  20  21            BRA     EVLCHK    ; AND TO EVLCHK WHEN WE GET BACK
  218  0136  7E  0000G     1$: JMP     BUG16A    ; THIS JUMPS TO THE PATCH SPACE
  219                                            ; TO FIX BUG NUMBER 16
  220  0139  DE  00G      NONEED: LDX  R2,D
  221  013B  20  02            BRA     KRAM
  222  013D  DE  00G      RESTRT: LDX  R6,D
  223  013F  DF  00G      KRAM: STX    R4,D
  224  0141  7E  0074'         JMP     ITEMS
  225  0144                BUG16B:
  226  0144  A6  04       PTR2: LDA A  4,X       ; UPDATE A POINTER TO WITHIN
  227  0146  E6  05            LDA B   5,X       ; AN ITEM
  228  0148  D0  01G           SUB B   R4+1,D    ; X+4 POINTS TO THE OLD POINTER
  229  014A  92  00G           SBC A   R4,D      ; TO WITHIN THE ITEM
  230                                            ; R4+1 POINTS TO WHERE THE ITEM
  231                                            ; USED TO START
  232  014C  D6  01G           ADD B   R3+1,D    ; AND R3 POINTS TO WHERE THE ITEM
  233  014E  99  00G           ADC A   R3,D      ; NOW STARTS
  234  0150  A7  04            STA A   4,X       ; SO (X+4)-(R4)-1+(R3, IS WHERE
  235  0152  E7  05            STA B   5,X       ; THE ITEM IS NOW
  236  0154  A6  01            LDA A   1,X
  237  0156  39                RTS
  238  0157                BUG16C:
  239  0157  DE  00G      EVLCHK: LDX  R0,D
  240  0159  81  00G           CMP A   ESTG,I    ; IF THE TAG ON THE STACK
  241  015B  26  0E            BNE     FIXUP     ; IS AN EVALUATOR STATUS
  242  015D  EE  04            LDX     4,X       ; CHECK THE NEXT LINE
  243  015F  9C  00G           CPX     R4,D      ; POINTER TO SEE IF WE NEED TO
```

```
  244  0161  26  08          BNE   FIXUP          ; UPDATE IT, IF NOT GO TO GOAHN
  245                                              ; IF SO, DO SO
  246  0163  96  00G         LDA A R3,D
  247  0165  A7  04          STA A 4,X
  248  0167  96  01G         LDA A R3+1,D
  249  0169  A7  05          STA A 5,X
  250  016B  8D  00000G      FIXUP: JSR  BACKUP
  251  016E  20  90          BRA   LOOP
  252  0170  DE  00G         NMTPTR: LDX  R3,D      ; UPDATE NT PTR FOR
  253  0172  EE  03          LDX   3,X              ; STRING OR ARRAY ELEMENT
  254  0174  96  00G         LDA A R3,D
  255  0176  A7  08          STA A 11.,X
  256  0178  96  01G         LDA A R3+1,D
  257  017A  A7  0C          STA A 12.,X
  258  017C  C5  60          BIT B  #60,I  ; IS THIS A STRING
  259  017E  26  B0          BNE   RESTRT           ; IF SO, NEED DO NO MORE UPDATING
  260                                              ; OR STACK SCANNING (SINCE MEM. COMPR
  261                                              ; DOESN'T RUN WHEN THERE IS A PTR
  262                                              ; TO THE ACTUAL STRING ON THE STACK
  263  0180  86  00G         LDA A #BEIG.I
  264  0182  9F  00G         STS   R0,D
  265  0184  20  03          BRA   RESCAN
  266  0186  BD  00000G      NXTAE: JSR  BACKUP
  267  0189  BD  00000G      RESCAN: JSR  LOCTG     ; IF THE ITEM WE FOUND IS AN
  268  018C  9C  00G         CPX   ZX,D             ; ARRAY SCAN THE STACK FOR
  269  018E  27  A0          BEQ   RESTRT           ; POINTERS TO ARRAY ELEMENTS
  270  0190  EE  02          LDX   2,X
  271  0192  9C  00G         CPX   R4,D
  272  0194  26  F0          BNE   NXTAE
  273  0196  DE  00G         LDX   R0,D             ; IF WE FOUND ONE UPDATE
  274  0198  96  01G         LDA A R3+1,D           ; BOTH THE ARRAY POINTER
  275  019A  A7  03          STA A 3,X
  276  019C  96  00G         LDA A R3,D
  277  019E  A7  02          STA A 2,X
  278  01A0  BD  A2          BSR   PTR2             ; AND THE ARRAY ELEMENT POINTER
  279  01A2  20  E2          BRA   NXTAE
  280  01A4  30              DONE: TSX              ; IF WE'RE DONE, CHECK WHETHER
  281  01A5  A6  04          LDA A 4,X              ; OR NOT WE RECOVERED ENOUGH
  282  01A7  E6  05          LDA B 5,X              ; SPACE
  283  01A9  D0  01G         SUB B LSP+1,D
  284  01AB  92  00G         SBC A LSP,D
  285  01AD  DB  01G         ADD B R2+1,D
  286  01AF  99  00G         ADC A R2,D
]]]]]
  287  01B1  DE  00G         LDX   R2,D
  288  01B3  DF  00G         STX   LSP,D
  289  01B5  40              TST A
  290  01B6  2D  0A          BLT   WIN              ; IF THIS IS NEGATIVE OR ZERO WE SUCCEEDED
  291  01B8  2E  04          BGT   LOSE
  292  01BA  C1  00          CMP B 0,I
  293  01BC  27  04          BEQ   WIN
  294  01BE  86  00G         LOSE:  LDA A ERWSFL,I  ; ELSE WE FAILED SO SET MEMORY
  295  01C0  97  00G         STA A ERRCD,D          ; FULL ERROR
  296  01C2  30              WIN:  TSX
  297  01C3  EE  01          LDX   1,X
  298  01C5  32              PUL A
  299  01C6  32              PUL A
  300  01C7  32              PUL A
```

```
301  01C8  32                    PUL A
302  01C9  32                    PUL A
303  01CA  32                    PUL A
304  01CB  DF  00G               STX     R0,D
305  01CD  BD  00C0G             JSR     ENABLE          ;CLEAR NOT REALLY
306  01D0  DE  00G               LDX     R0,D
307  01D2  6E  G0                JMP     0,X
308  01D4  DF  00G       ADDLEN: STX     R4,D            ;X POINTS TO AN ITEM
309  01D6  96  01G               LDA A   R4+1,D          ;AND WE WANT TO ADVANCE
310  01D8  AB  01                ADD A   1,X             ;IT TO THE NEXT ITEM
311  01DA  97  01G               STA A   R4+1,D          ;SO ADD THE LENGTH (AT X+0
312  01DC  96  00G               LDA A   R4,D            ;AND X+1 TO IT)
313  01DE  A9  00                ADC A   0,X
314  01E0  97  00G               STA A   R4,D
315  01E2  DE  00G               LDX     R4,D
316  01E4  39                    RTS
317         0001'               .END
```

```
COMPR  RT-11 MMAC  VM02-10   14-OCT-76 01:29:41 PAGE 1+
SYMBOL TABLE

ADDLEN   01D4R       AESCAN   0189R       BACKUP= XXXXXX G       BUG16A= XXXXXX G       BUG16B   0144RG
BUG16C   0157RG      CDOPTR= XXXXXX G     CDSPTR= XXXXXX G       CLPTR = XXXXXX G       COMPR =  0000RG
DISBLE= XXXXXX G     DONE     01A4R       ENABLE= XXXXXX G       ERRCD = XXXXXX G       ERWSFL= XXXXXX G
ESTG  = XXXXXX G     EVLCHK   0157R       FIXUP    0168R         FORTG = XXXXXX G       GARBGE   0071R
GOAWN    00E0R       GOSTG = XXXXXX G     IMXTG = XXXXXX G       IN       007BR         ITEMS    0074R
KBFLAG= XXXXXX G     KRAM     013FR       KWIK     00A2R         LISTTG= XXXXXX G       LOCTG = XXXXXX G
LOCTGR= XXXXXX G     LOOP     0100R       LOSE     018ER         LSP   = XXXXXX G       NEXT     0026R
NLPTR = XXXXXX G     NMIPTR   0170R       NONEED   0139R         NTPTR = XXXXXX G       NXTAE    0186R
NXTPGM   0044R       PAETG = XXXXXX G     PGMPTR= XXXXXX G       PGMTG = XXXXXX G       PLOOP    0038R
PLOSTG= XXXXXX G     PLSTRT   0036R       PRTTG = XXXXXX G       PSTRT    002AR         PTR2     0144R
RESTRT   013DR       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
SAFE  = XXXXXX G     SEARCH   0064R       SKIPNG   006FR         START    0056R         STPTR = XXXXXX G
USRORG= XXXXXX G     WLOOP    0008R       WIN      01C2R         ZX    = XXXXXX G

  ABS.    0000     00
          01E5     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3192. WORDS
.SY:COMPR/C<DK1:SEICLI,COMPR
```

CROSS REFERENCE TABLE (CREF V01-03)

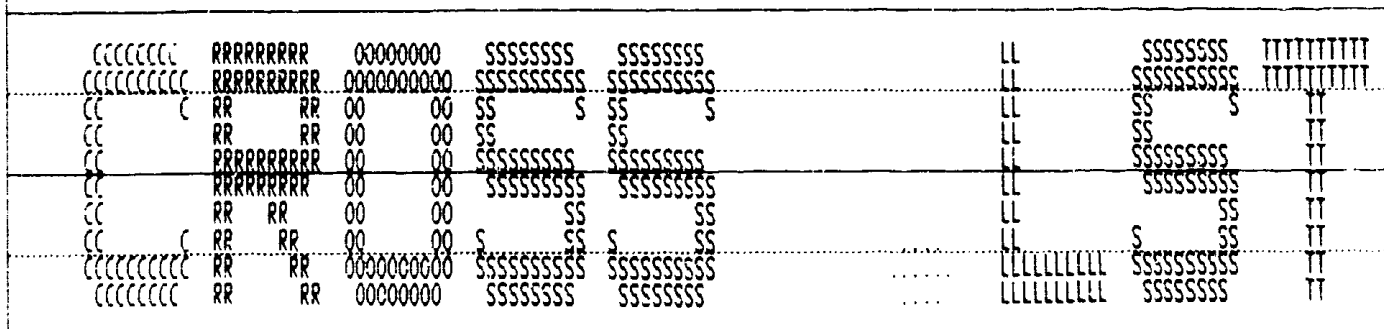
	1-18														
ARGLEN	1-114	1-117	1-308#												
BESCAN	1-265	1-267#													
BACKUP	1-22#	1-98	1-250	1-266											
BUG16A	1-20#	1-218													
BUG16B	1-20#	1-225#													
BUG16C	1-20#	1-238#													
CDOPTR	1-26#	1-183													
CDSPTR	1-26#	1-179	1-181*	1-182*											
CLPTR	1-22#	1-81	1-187	1-189*	1-190*										
COMPR	1-17#	1-18#													
DISBLE	1-25#	1-60													
DONE	1-110	1-120	1-280#												
ENABLE	1-25#	1-305													
ERRCD	1-22#	1-295*													
ERUSEL	1-22#	1-294													
ESTG	1-24#	1-212	1-240												
EVLCHK	1-205	1-215	1-217	1-239#											
FIXUP	1-241	1-244	1-250#												
FORTG	1-24#														
GARBGE	1-117#	1-122													
GOBUK	1-170	1-176#	1-185												
GOSTG	1-24#														
IMXTG	1-24#	1-198													
IN	1-109	1-119	1-121#												
ITEMS	1-118#	1-224													
KBFLAG	1-23#														
KRAM	1-221	1-223#													
KWIK	1-145#	1-154													
LISTTG	1-24#														
LOCTG	1-23#	1-95	1-267												
LOCTGR	1-25#	1-199													
LOOP	1-194	1-197#	1-251												
LOSE	1-291	1-294#													
LSP	1-22#	1-108	1-113	1-283	1-284	1-288*									
NEXT	1-64	1-66	1-77#												
NLPTR	1-26#	1-193	1-195*	1-196*											
NMTPTR	1-157	1-252#													
NONEED	1-140	1-220#													
NTPTR	1-26#	1-191													
NXTAE	1-266#	1-272	1-279												
NXTPGM	1-87	1-94#													
PAETG	1-25#	1-263													
PGMPTR	1-23#	1-86	1-168	1-171											
PGMTG	1-23#	1-94													
PLOOP	1-87#	1-93	1-101												
PLOSTG	1-24#	1-197	1-214												
PLSTRT	1-82	1-86#													
PRTTG	1-25#														
PSTRT	1-62	1-80#													
PTR2	1-184	1-192	1-216	1-226#	1-278										
R0	1-21#	1-67*	1-72	1-74	1-76	1-80*	1-142*	1-207	1-239	1-264*	1-273	1-304*	1-306		
R1	1-21#														
R2	1-21#	1-97*	1-99	1-106*	1-129	1-133	1-134*	1-136	1-137*	1-153	1-220	1-285	1-286	1-287	
R3	1-21#	1-170*	1-178	1-194	1-158	1-161	1-163	1-165	1-172	1-174	1-176	1-177	1-208	1-210	
	1-232	1-233	1-246	1-248	1-252	1-254	1-256	1-274	1-276						

CROSS REFERENCE TABLE (CREF V01-03)

R4	1-21#	1-105#	1-116#	1-131	1-169	1-178	1-204	1-223#	1-228	1-229	1-243	1-271	1-308#	1-309
	1-311#	1-312	1-314#	1-315										
R5	1-21#	1-145#	1-150											
R6	1-21#	1-143#	1-146	1-149#	1-222									
R7	1-21#													
RESTRT	1-201	1-222#	1-259	1-269										
RTRNTG	1-23#	1-58												
SAFE	1-25#													
SEARCH	1-111#	1-115												
SKIPNG	1-112	1-116#												
START	1-96	1-103#												
STPTR	1-23#	1-61												
USRORG	1-20#	1-103												
VLOOP	1-62#	1-78												
WIN	1-290	1-293	1-296#											
ZX	1-20#	1-200	1-268											

CROSS REFERENCE TABLE (CREF V01-03)

SEQ 1-36



```
                                .TITLE  CROSS    DRAW CROSS ON SCREEN AND INPUT POINTS.
                                .IDENT  /JKC010/
                                .GLOBL  QCROSS,CROS
                                .GLOBL  TMPHX,TMPLX,TMPLY,TMPHY
                                .GLOBL  P1AHX,P1ALX,P1AHY,P1ALY
                                .GLOBL  P1AMTA
                                .GLOBL  CURSOR,BLINK
                                .GLOBL  QIN,QOUT,KBQOUT
                                .GLOBL  GENCUR,GINSET,A7X,ADRDEV,OPRADR,DIMSTR
                                .GLOBL  EOLTG,ZGIN,INT1,ERRCD
                        0020    XCOMP=40
                        0080    YCOMP=200
```

```
    1  0000  86  00G       QCROSS: LDA A  EOLTG,I      ; SET UP STACK FOR GIN.
    2  0002  36                    PSH A
    3  0003  30                    TSX                 ; GET TABLE POINTER.
    4  0004  EE  04                LDX    4,X
    5  0006  A6  04                LDA A  4,X          ; GET ATTRIBUTE.
    6  0008  81  10                CMP A  20,I         ; DIMENSIONED ?
    7  000A  27  10                BEQ    CS.1         ; YES, DON'T REDIMENSION.
    8  000C  7F  0000G             CLR    INT1         ; NO, DO DEFAULT DIMENSION.
    9  000F  86  48                LDA A  72.,I
   10  0011  97  01G               STA A  INT1+1,D
   11  0013  BD  0000G             JSR    DIMSTR
   12  0016  96  00G               LDA A  ERRCD,D      ; GET ERROR CODE.
   13  0018  27  02                BEQ    CS.1         ; CONTINUE IF NO ERRORS.
   14  001A  32                    PUL A               ; SCRATCH TAG.
   15  001B  39                    RTS                 ; ERROR EXIT.
   16  001C  8D  02        CS.1:   BSR    CROS         ; DRAW "CROSS" ON SCREEN.
   17  001E  20  22                BRA    CRCNT        ; FINISH.
   18  0020  CE  0200      CROS:   LDX    512.,I       ; FORCE TO SCREEN CENTER.
   19  0023  DF  00G               STX    TMPHX,D
   20  0025  CE  0186              LDX    390.,I
   21  0028  DF  00G               STX    TMPHY,D
   22  002A  B6  0000G             LDA A  PIAHX        ; GET VECTOR CONTROL.
   23  002D  8A  40                ORA A  100,I        ; SET VECTOR OFF.
   24  002F  B7  0000G             STA A  PIAHX        ; RESTORE FOR CONTROL.
   25  0032  86  81                LDA A  129.,I       ; GET GRAPHICS POINTER.
   26  0034  97  00G               STA A  CURSOR,D     ; MAKE DISPLAY CHARACTER
   27  0036  DE  00G       CROSLP: LDX    TMPHY,D      ; RESTORE Y D/A VALUE.
   28  0038  BD  00C1'             JSR    SET          ; SET UP D/A'S.
   29  003B  85  20                BIT A  YCOMP,I      ; EXTRACT Y POSITION.
   30  003D  26  10                BNE    INCY         ; Y D/A BELOW JOY.
   31  003F  8C  0000      DECY:   CPX    0,I          ; AT BOTTOM ?
   32  0042  27  19                BEQ    TESTX        ; YES, CHECK X.
   33  0044  09                    DEX                 ; NO, MOVE DOWN ONE.
   34  0045  DF  00G               STX    TMPHY,D      ; SAVE FOR MOVE.
   35  0047  8D  78                BSR    SET          ; STROBE IN NEW VALUE.
   36  0049  85  80                BIT A  YCOMP,I      ; EXTRACT Y POSITION.
   37  004B  27  F2                BEQ    DECY         ; NOT THERE YET.
   38  004D  20  0E                BRA    TESTX        ; Y OK, NOW SLEW X.
   39  004F  8C  030C      INCY:   CPX    780.,I       ; AT TOP ?
   40  0052  27  09                BEQ    TESTX        ; YES, GO SLEW X.
   41  0054  08                    INX                 ; NO, MOVE UP 1.
   42  0055  DF  00G               STX    TMPHY,D      ; SAVE FOR MOVE.
   43  0057  8D  68                BSR    SET          ; STROBE IN.
   44  0059  85  80                BIT A  YCOMP,I      ; EXTRACT Y POSITION.
   45  005B  26  F2                BNE    INCY         ; NOT THERE YET.
   46  005D  DE  00G       TESTX:  LDX    TMPHX,D      ; X = X D/A POSITION.
   47  005F  85  20                BIT A  XCOMP,I      ; EXTRACT X POSITION.
   48  0061  27  10                BEQ    INCX         ; X D/A LEFT OF JOY.
   49  0063  8C  0000      DECX:   CPX    0,I          ; AT LEFT ?
   50  0066  27  19                BEQ    CRSDRW       ; YES, GO DRAW CURSER.
   51  0068  09                    DEX                 ; NO, MOVE LEFT ONE.
   52  0069  DF  00G               STX    TMPHX,D      ; SAVE FOR MOVE.
   53  006B  8D  54                BSR    SET          ; STROBE IN NEW VALUE.
   54  006D  85  20                BIT A  XCOMP,I      ; EXTRACT X POSITION.
   55  006F  26  F2                BNE    DECX         ; NOT THERE YET.
   56  0071  20  0E                BRA    CRSDRW       ; X OK, DRAW CURSOR.
   57  0073  8C  03FF      INCX:   CPX    1023.,I      ; AT RIGHT EDGE ?
```

```
                0076  27  09            BEQ     CRSDRW          ; YES, GO DRAW CURSOR.
                0078  08                INX                     ; NO, MOVE RIGHT 1.
                0079  DF  00G           STX     TMPHX,D         ; SAVE FOR MOVE.
                007B  8D  44            BSR     SET             ; STROBE IN NEW VALUE.
                007D  85  20            BIT A   XCOMP,I         ; EXTRACT X POSITION.
                007F  27  F2            BEQ     INCX            ; NOT THERE YET.
                0081  BD  0000G CRSDRW: JSR     GENCUR          ; DRAW CURSOR.
                0084  DE  00G   TRYAGN: LDX     QIN,D           ; CHECK FOR CHARACTER INPUT
                0086  9C  00G           CPX     QOUT,D
                0088  27  AC            BEQ     CROSLP          ; NONE THERE, CONTINUE.
                008A  BD  0000G         JSR     KBQOUT          ; GET CHARACTER.
                008D  81  9B            CMP A   155.,I          ; CHECK FOR CONTROL KEY CODES.
                008F  22  F3            BHI     TRYAGN          ; IGNORE CONTROL KEYS.
                0091  84  7F            AND A   177,I           ; MASK OFF NUMED POINTER
                0093  7F  0000G         CLR     BLINK           ; MAKE SURE NO REFRESH PRINTING TAKES PLACE.
                0096  39                RTS                     ; RETURN.
                0097  81  0D    CRCNT:  CMP A   15.I            ; IS IT CR ?
                0099  26  03            BNE     SET1            ; NO, CHARACTER COUNT = 1.
                009B  5F                CLR B                   ; YES, MAKE NULL STRING.
                009C  20  02            BRA     POINT           ; CONTINUE
                009E  C6  01    SET1:   LDA B   1,I             ; CHARACTER COUNT = 1.
                00A0  30        POINT:  TSX                     ; POINT TO TABLE POINTER.
                00A1  EE  04            LDX     4,X             ; GET TABLE POINTER.
                00A3                    SEI                     ; DISABLE INTERRUPTS.
          00A3  01  0F          .BYTE   01,17
                00A5  6F  07            CLR     7,X             ; SET CHARACTER COUNT = 1.
                00A7  E7  08            STA B   8.,X
                00A9  EE  0B            LDX     11.,X           ; GET DATA POINTER.
                00AB  A7  05            STA A   5,X             ; STORE CHARACTER.
                00AD  86  20            LDA A   40,I            ; GET DEFINED BIT.
                00AF  A7  02            STA A   2,X             ; SET DEFINED ATTRIBUTE.
                00B1  0E                CLI                     ; ENABLE INTERRUPTS.
                00B2  CE  0C00G         LDX     ZGIN,I          ; SET UP INTERNAL HARDWARE REFERENCE.
                00B5  DF  00G           STX     OPRADR,D
                00B7  BD  0000G         JSR     ADRDEV
                00BA  30                TSX                     ; GET PRESENT STACK POINTER.
                00BB  BD  0000G         JSR     A?X             ; POINT TO "Y".
                00BE  7E  0000G         JMP     GINSET          ; LET GIN FINISH CROSS.
```

```
00C1   96    00G          SET:    LDA A   TMPHY,D      ; SET HIGH Y.
00C3   B7  0000G                  STA A   PIAHY
00C6   96    00G                  LDA A   TMPLY,D      ; SET LOW Y.
00C8   B7  0000G                  STA A   PIALY
00CB   B6  0000G                  LDA A   PIAHX        ; MASK OFF OLD COUNT AND PRESERVE OLD BITS.
00CE   84    FC                   AND A   374,I
00D0   9B    00G                  ADD A   TMPHX,D      ; SET IN HIGH ORDER COUNT.
00D2   B7  0000G                  STA A   PIAHX        ; SET HIGH X.
00D5   96    00G                  LDA A   TMPLX,D      ; SET LOW X AND STROBE VALUES INTO D/A'S.
00D7   B7  0000G                  STA A   PIALX
00DA   C6    06                   LDA B   6,I          ; GET DELAY COUNT.
00DC   5A                 DELAY:  DEC B                ; SETTLING TIME DELAY.
00DD   26    FD                   BNE     DELAY
00DF   B6  0000G                  LDA A   PIAMTA       ; GET X,Y COMPARATOR READING
00E2   39                         RTS                  ; RETURN.
      0001'                       .END
```

```
CROSS   DRAW CROSS ON SCREEN A       RT-11 MMAC  VM02-10   14-OCT-76 01:29:50 PAGE 3+
SYMBOL TABLE

ADRDEV= XXXXXX G      A7X   = XXXXXX G      BLINK = XXXXXX G      CRCNT    0097R      CROS     0020RG
CROSLP    0036R       CRSDRW    0081R       CS.1      001CR       CURSOR= XXXXXX G      DECX     0063R
DECY      003FR       DELAY     000CR       DIMST%= XXXXXX G      EOLTG = XXXXXX G      ERRCD = XXXXXX G
GENCUR= XXXXXX G      GINSET= XXXXXX G      INCX      0073R       INCY     004FR      INT1  = XXXXXX G
KBQOUT= XXXXXX G      OPRADR= XXXXXX G      PIAPX = XXXXXX G      PIAPY = XXXXXX G      PIALX = XXXXXX G
PIALY = XXXXXX G      PIAMTN= XXXXXX G      POINT     00A0R       QCROSS   0000RG     QIN   = XXXXXX G
QOUT  = XXXXXX G      SET       00C1R       SET1      009ER       TESTX    0050R      TMPHX = XXXXXX G
TMPHY = XXXXXX G      TMPLX = XXXXXX G      TMPLY = XXXXXX G      TRYAGN   0084R      XCOMP =   0020
YCOMP =   0080        ZGIN  = XXXXXX G
. ABS.    0000      00
          00E3      01

ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3312. WORDS
.SY:CROSS/C(DK1:SETCLI,CROSS
```

CROSS REFERENCE TABLE (CREF V01-03)

```
AZX       1-24#   2-93
ADRDEV    1-24#   2-91
BLINK     1-22#   2-72*
CRCNT     2-17    2-74#
CROS      1-18#   2-16    2-18#
CROSLP    2-27#   2-67
CRSDRW    2-50    2-56    2-58    2-64#
CS.1      2-7     2-13    2-16#
CURSOR    1-22#   2-26*
DECX      2-49#   2-55
DECY      2-31#   2-37
DELAY     3-12#   3-13
DIMSTR    1-24#   2-11
EOLTG     1-25#   2-1
ERRCD     1-25#   2-12
GENCUR    1-24#   2-64
GINSET    1-24#   2-94
INCX      2-48    2-57#   2-63
INCY      2-30    2-39#   2-45
INT1      1-25#   2-8*    2-10*
KBQOUT    1-23#   2-68
OPRADR    1-24#   2-90*
PIAHX     1-20#   2-22    2-24*   3-5     3-8*
PIAHY     1-20#   3-2*
PIALX     1-20#   3-10*
PIALY     1-20#   3-4*
PIAMTA    1-21#   3-14
POINT     2-27    2-29#
QCROSS    1-18#   2-1#
QIN       1-23#   2-65
QOUT      1-23#   2-66
SET       2-28    2-35    2-43    2-53    2-61    3-1#
SET1      2-75    2-78#
TESTX     2-32    2-38    2-40    2-46#
TMPHX     1-19#   2-19*   2-46    2-52*   2-60*   3-7
TMPHY     1-19#   2-21*   2-27    2-34*   2-42*   3-1
TMPLX     1-19#   3-9
TMPLY     1-19#   3-3
TRYAGN    2-65#   2-70
XCOMP     1-26#   2-47    2-54    2-62
YCOMP     1-27#   2-29    2-36    2-44
ZGIN      1-25#   2-89
```

CROSS REFERENCE TABLE (CREF V01-03)

SCI 1-3# 2-81

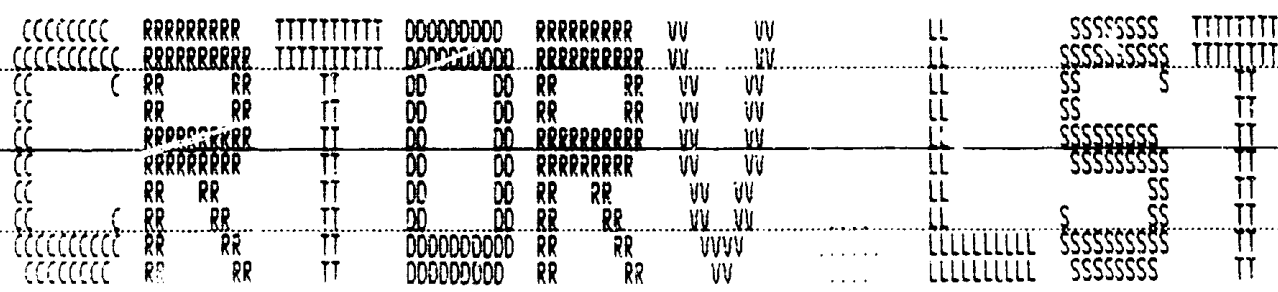


TABLE OF CONTENTS

```
                                .TITLE  CRTDRV  DISPLAY DRIVER
                                .IDENT  /RM0043/
                                .ENABL LC
                                .GLOBL  PIAHY,PIALY,PIAHX,PIALX,PIALT
                                .GLOBL  LINELN
                                .GLOBL  FULL,BLINK,DOTON,COLCT,A6X,FONT
                                .GLOBL DSPAGE,VECTOR,CHSCAN,SETBNK,CHRVCT
                                .GLOBL TMPLY,TMPHY,TMPLX,TMPHX,PCHAR,PCHAR2
                                .GLOBL TMP1,TMP2,TMP3
                                .GLOBL  DSPCP2,BANK
                                .GLOBL  CRTDRV
0000'                           CRTDRV  =.


                        ; xxxxxxxxxx  SEE DSPEML  FOR GEROGE'S CHANGES xxxx


                        ;
                        ;THE FOLLOWING IS THE INIT ROUTINE FOR THE DISPLAY SECTION OF THE TEK 51
                        ; BE CAREFUL OF INTERACTION BETWEEN CRT AND THE TAPE READER
                        ; WHEN THE DISPLAY IS ON THE BIT IN TAPE PIA MUST BE HIGH TO
                        ; DISCONNECT A 8 BIT SHIFT REGISTER AND PUT IT IN THE TRI-STATE
                        ; MODE-THIS IS AT LOCATION 103632---IT SHOULD CONTAIN A 17 OCTAL
                        ; TO GUARANTEE THAT THE TAPE READER DOES NOT SEE FLUX
                        ; REVERSALS THAT COULD DESTROY DATA ON THE TAPE
                        ; THIS CAN BE DONE BY INITING THE TAPE FIRST AND THEN THE
                        ; CRT.  SEE IWATA--

                        ; WHEN THE TAPE IS RUNNING THE 3RD LSB OF PIA AT 103632 MUST
                        ; BE AT A LOGIC '0' TO DISCONNECT THE DISPLAY CIRCUIT FROM
                        ; HIGH Y AND THEN THE PIA MUST BE INITED TO BE A READ IN
                        ; ONLY DEVICE--PRIOR TO THIS THE STATUS OF THE DATA LOCATED
                        ; IN GLOBALS PIAHY AND PIALY MUST BE SAVED AND THE DISPLAY
                        ; BROUGHT TO THE CENTER OF THE SCREEN ANYTIME THAT THE TAPE IS
                        ; CALLED--THIS IS VERY IMPORTANT AS THE DISPLAY DRAWS VERY
                        ; LITTLE UNREG POWER AT THE CENTER OF THE SCREEN

                        ; SEE IWATA FOR HOW HE WANTS OUR SHARED PIA INITED

                        ; THE CENTER SCREEN COORDINATES ARE 512 X --
                        ; AND 390 Y--DECIMAL COORDINATES. THE FOLLOWING
                        ; SMALL PROGRAM WILL CENTER THE SCREEN

                        ;       LDA A   206,I
                        ;       STA A   PIALY           LOW 8 BITS OF Y
                        ;       LDA A   1,I
                        ;       STA A   PIAHY             2 LSB OF PIA BUT MSB OF Y
                        ;       LDA A   2,I
                        ;       ORA A   374,I
                        ;       STA A   PIAHX
                        ;       CLR     PIALX


                        ; ANY WRITE TO PIALX CAUSES DATA TO BE STROBED INTO THE
                        ; 20 BITS OF LATCH THAT HOLD THE DISPLAY POSITION
                        ; THEREFORE PIALX MUST ALWAYS BE DONE LAST!!!!!!
```

```
331
332                             ; PRIOR TO MOVING THE DISPLAY TO CENTER DON'T FORGET
333                             ; TO SAVE ITS STATUS SOMEWHERE--LEAVE MY SOMEWHERES ALONE!!
334                             ;
```

```
                                        .SBTTL  MAIN DISPLAY DRIVER
                                        .GLOBL  DRBUSY
                                ;THIS ROUTINE WAITS UNTIL DRBUSY AND VECTOR ARE IN THE READY
                                ;STATE
                                ;
0000    F6  0000G   DRBUSY: LDA B   PIAHX
0003    C8  04              EOR B   #4,I
0005    C5  0C              BIT B   #14.,I
0007    26  F7              BNE     DRBUSY
0009    39                  RTS
                                ;
                                ;*****************************************************************
                                ;       PAGE, HOME AND CR
                                ;
000A                    DSPAGE:
000A    86  EC      PAGE:   LDA A   #354,I          ;CLEARS THE CRT AND HOMES THE CURSOR
000C    B7  0000G           STA A   PIAHX           ;TURNS ON EXT PAGE COMMAND
000F    8A  10              ORA A   #20,I
0011    B7  0000G           STA A   PIAHX           ;TURN OFF PAGE COMMAND
                            .GLOBL  DSHOME
0014                    HOME:
0014    CE  02FF    DSHOME: LDX     #767.,I         ; MOVE TO REAL HOME TO MAKE EVERY ONE HAPPY
0017    DF  00G             STX     TMPHY.D
0019    01                  NOP
001A    07                  TPA                     ; WELL IT'S NON-ZERO
001B    97  00G             STA A   TMP1.D
001D    BD  04ED'           JSR     MOVE
0020    7F  0000G           CLR     FULL
0023    01          CR:     NOP
0024    01                  NOP
0025    86  FC              LDA A   #374,I          ; SEND DOT TO ZERO X
0027    B7  0000G           STA A   PIAHX
002A    4F                  CLR A
002B    B7  0000G           STA A   PIALX
002E    97  00G             STA A   TMPHX.D         ; UPDATE TEMP REG
0030    97  00G             STA A   TMPLX.D
                                ;
                                ;*****************************************************************
                                ;       SOME DISPLAY SERVICE ROUTINES
                                ;
                                .GLOBL  WAIT
                                ;THIS ROUTINE USES THE MAGTAPE 40US ONE SHOT (PIAHX CB2 TRIGGERS IT)
                                ;TO GENERATE A WAIT OF 10.0 MS
0032    86  FA      WAIT:   LDA A   #250.,I         ; SET FOR 10MS
0034    7E  0000G           JMP     DELY1S
                            .GLOBL  DELY1S
0037    81  0A      NXTIST: CMP A   #12.,I          ;LINE FEED ?
0039    26  26              BNE     THOME                   ;BRANCH TO TEST FOR HOME COMMAND NEXT
003B    96  00G     LF:     LDA A   TMPHY.D                 ;LINE FEED ROUTINE FOLLOWS
003D    26  09              BNE     3$
003F    D6  00G             LDA B   TMPLY.D
0041    C0  16              SUB B   #22.,I
0043    24  10              BCC     PUPD                    ;PUPD STANDS FOR PARTIAL DISPLAY UPDATE
0045    D7  00G             STA B   FULL.D
0047    39                  RTS                     ;RETURN TO CALLING PLACE WITHOUT
                                                    ;CHANGING Y POSITION DATA-PAGE IS
                                                    ;FULL AND FLAG IS SET
```

```
    1  0048  D6  0GG        3$:  LDA B  TMPLY.D
    2  004A  C0  16             SUB B  22.I
    3  004C  29  02             BVS    UPDATE          ;FULL DISPLAY UPDATE
    4  004E  82  00             SBC A  0.I
    5  0050  97  0GG       UPDATE: STA A  TMPHY.D
    6  0052  B7  0000G            STA A  PIAHY
    7  0055  D7  0GG        PUPD: STA B  TMPLY.D
    8  0057  F7  0000G            STA B  PIALY
    9  005A  B6  0000G            LDA A  PIALX
   10  005D  B7  0000G            STA A  PIALX
   11  0060  39                   RTS
   12  0061  81  1E         THOME: CMP A  36.I            ;HOME ?
   13  0063  26  03             BNE    5$
   14  0065  7E  0014'           JMP    HOME
   15  0068  81  0C         5$: CMP A  14.I             ;PAGE ?
   16  006A  26  03             BNE    6$
   17  006C  7E  000A'           JMP    PAGE
   18  006F  81  07         6$: CMP A  7.I              ;BELL ?
   19  0071  26  24             BNE    TSTBS
   20                            .GLOBL BELCAL
   21  0073  B6  0000G      BELCAL: LDA A  PIALT
   22  0076  36                   PSH A
   23  0077  5F                   CLR B                  ; DUTY CYCLE MODULATION OF SPEAKER
   24  0078  86  46               LDA A  70.,I
   25  007A  36              3$:  PSH A
   26  007B  08              1$:  INX
   27  007C  09                   DEX
   28  007D  4A                   DEC A
   29  007E  26  FB               BNE    1$
   30  0080  B6  0000G            LDA A  PIALT
   31  0083  38  80               EOR A  200.I           ; TOGGLE BELL
   32  0085  B7  0000G            STA A  PIALT
   33  0088  2B  04               BMI    2$
   34  008A  5C                   INC B
   35  008B  17                   TBA
   36  008C  20  ED               BRA    1$
   37  008E  32              2$:  PUL A
   38  008F  4A                   DEC A
   39  0090  26  E8               BNE    3$
   40  0092  32                   PUL A
   41  0093  B7  0000G            STA A  PIALT
   42  0096  39                   RTS
```

```
                                        ;LOOK FOR BACKSPACE
                                        ;
0097  81  08    TSTBS:  CMP A   10,I     ;BACK SPACE ?
0099  26  19            BNE     8$
0098  D6  00G           LDA B   TMPLX.D  ;LOW X DATA ON POSITION
009D  96  00G           LDA A   TMPHX.D  ;HIGH DATA X POSITION
009F  26  04            BNE     41$      ;IF TRUE DO SUBTRACT ROUTINE
00A1  C1  0E            CMP B   16.I     ;HI X=0 ENOUGH ROOM IN LO X ?
00A3  25  06            BCS     42$      ;FOLD AROUND IF TRUE
00A5  C0  0E    41$:    SUB B   16.I     ;SUBTRACT ROUTINE FOR BACK SPACE
00A7  82  00            SBC A   0.I      ;POSSIBLE NEW HI X VALUE
00A9  20  46            BRA     GOUPX
00AB  86  03    42$:    LDA A   3.I      ;HI X FOLD AROUND POSITION
00AD  C6  E2            LDA B   342.I    ;LO X "" "" ""
00AF  8D  40            BSR     GOUPX    ; DO IT
0081  7E  0032'         JMP     WAIT.    ; WAIT FOR SCREEN TO STALBLIZE
                                        ;
                                        ;LOOK FOR A VERICAL TAB
                                        ;
00B4  81  0B    8$:     CMP A   13.I     ;VERTICAL TAB ?
00B6  26  1D            BNE     9$
00B8  96  00G           LDA A   TMPHY.D  ;VERTICAL TAB ROUTINE
00BA  D6  00G           LDA B   TMPLY.D
00BC  81  03            CMP A   3.I
00BE  27  10            BEQ     63$      ;HI Y =3 DO FOLDOVER ROUTINE
00C0  4F                CLR A
00C1  CB  16            ADD B   22.,I
00C3  99  00G           ADC A   TMPHY.D
00C5  81  03            CMP A   3.I
00C7  27  04            BEQ     62$      ;GO DO TST LOY FOR ZERO
00C9  7E  00E3  61$:    JMP     UPDATE
00CC  39        50$:    RTS
00CD  50        62$:    TST B
00CE  27  F9            BEQ     61$
00D0  C6  14    63$:    LDA B   20.,I    ;FOLDOVER RESET 1ST TAB
00D2  4F                CLR A
00D3  20  F4            BRA     61$
                                        ;
```

```
                                        ;LOOK FOR HORIZONTAL TAB
                                        ;
0005  81 09           9$:   CMP A  11,I       ;HORIZ TAB ?
0007  26 F3                 BNE    5$
0009  D6 00G                LDA B  TMPLX,D    ;LO X VALUE
000B  96 00G                LDA A  TMPHX,D    ;HI X VALUE NOW DO H TAB ROUTINE
000D  27 0C                 BEQ    CKTAB2     ;COULD BE TAB 2 OR TAB 3
000F  4A                    DEC A
0010  27 18                 BEQ    CKTAB4
0012  4A                    DEC A
0013  26 27                 BNE    CRLF       ;ITS AFTER THE LAST TAB
0015  C1 F4                 CMP B  364,I      ;DO TAB 4 OR CRLF
0017  25 15                 BCS    SET4       ;SET TAB 4 DATA
0019  20 21                 BRA    CRLF       ;GO TO NEXT LINE
001B  C1 FC           CKTAB2: CMP B  374,I    ;IF EQ OR GT SET NEXT HIGHER TAB
001D  24 05                 BCC    SET3       ;SET TAB 3 IF BRANCH TAKEN
001F  C6 FC           SET2:   LDA B  374,I    ;ACC A IS 0. THIS IS TAB 2 DATA
0021  7E 0277'        GOUPX:  JMP    UPDATX     ;UPDATE X POSITION AS PART OF SPACE ROUTINE
0024  86 01           SET3:   LDA A  1,I        ;TAB 3 DATA
0026  C6 F8                   LDA B  370,I
0028  20 F7                 BRA    GOUPX      ;UPDATE X POSITION
002A  C1 F8           CKTAB4: CMP B  370,I    ;TAB 3 OR TAB 4?
002C  25 F6                 BCS    SET3       ;TAB 3 IF BRANCH TAKEN
002E  86 02           SET4:   LDA A  2,I        ;TAB 4 DATA
0030  C6 F4                   LDA B  364,I
0032  20 ED                 BRA    GOUPX
0034  81 0D           PCHR.1: CMP A  15,I       ;IS IT CR
0036  27 04                 BEQ    CRLF       ; NO
0038  81 1F                 CMP A  31,I       ; ALTERNATE "CR" --> "
003A  26 05                 BNE    PCHARZ
003C  BD 0038'        CRLF:   JSR    LF         ;MAKE CR INTO LF AND CR
003F  86 0D                   LDA A  15,I       ;FIX UP ACC-A
0041  81 80           PCHARZ: CMP A  200,I      ; CHECK FOR BLOT
0043  26 06                 BNE    CKFONT
0045  CE 045C'        BLOTIT: LDX    BLOT,I
0048  7E 0104'                JMP    CHSCAN
```

```
                                ;THIS SECTION CHECKS FOR NON-US CHARACTER FONT.  IF ANY OF THE ALTERNATE
                                ;       CHARS ARE PRESENT AN ALTERNATE DOT PATTERN IS SUBSTITUTED.
                                ;
                                ; FONT = 0      US FONT
                                ; FONT = 1      SCANDINAVIAN FONT
                                ; FONT = 2      GERMAN FONT
                                ; FONT = 3      GENERAL EUROPEAN FONT (FRENCH, BRITISH, ITALIAN)
                                ; FONT = 4      SPANISH FONT
                                ;
                                ; FONT = 5      SPECIAL GRAPHIC CHARACTERS
                                ; FONT = 6      SPECIAL CHARACTERS (APL)
0118  D6  006G    CKFONT: LDA B   FONT,D          ;IS FONT FOR THE U.S. (=0) ?
011D  27  5A              BEQ     CKSPCE          ;DONE CHECK FOR A SPACE
011F  C1  05              CMP B   5,I             ; FONTS > 5 BECOME 0
0121  22  56              BHI     CKSPCE
0123  CE  0468'           LDX     NONUS,I         ;LOAD POINTER TO FIRST SET OF DOTS
0126  81  40              CMP A   '@,I            ;CHAR = @ ?
0128  27  3B              BEQ     CHSCB           ;YES
012A  09                  DEX                     ;DEC POINTER FOR JSR TO DSPCOM
012B  C1  05              CMP B   5,I             ;IS IT A SPECIAL GRAPHICS CHAR
012D  26  0C              BNE     1$              ;YES
012F  81  23              CMP A   '#,I
0131  27  E2              BEQ     BLOIIT
0133  8D  25              BSR     DSPCOM          ;CHAR =LEFT BRACE-MAP TO A LEFT ARROW
0135  8D  23              BSR     DSPCOM          ;CHAR =VERT BAR-MAP TO GRAPH POINT
0137  8D  21              BSR     DSPCOM          ;CHAR =RT BRACE-MAP TO RT ARROW
0139  20  3E              BRA     CKSPCE          ;DONE-CHECK SPACE
013B  CE  0479'     1$:   LDX     GNFONT-6,I      ;SET TABLE RIGHT
013E  8D  1A              BSR     DSPCOM          ;CHAR =#-MAP TO STERLING POUND
0140  8D  18              BSR     DSPCOM          ;CHAR =$-MAP TO INTER MONETARY SYMBOL
0142  C1  03              CMP B   3,I             ;GENERAL EUROPE FONT?
0144  27  33              BEQ     CKSPCE          ;YES CHECK FOR SPACE
0146  22  28              BHI     DSPSK2          ;SPANISH FONT
0148  8D  10              BSR     DSPCOM          ;CHAR =(-MAP TO (A) WITH 2 DOTS ABOVE
014A  8D  0E              BSR     DSPCOM          ;CHAR =LEFT BRACE-MAP TO (A)WITH 2 DOTS ABOVE
014C  8D  0C              BSR     DSPCOM          ;CHAR =BACKSLASH-MAP TO (O) WITH 2 DOTS ABOVE
014E  8D  0A              BSR     DSPCOM          ;CHAR =VERTICAL BAR-MAP TO (O) WITH 2 DOTS ABOVE
0150  C1  02              CMP B   2,I             ;GERMAN FONT ?
0152  27  13              BEQ     DSPSK1          ;YES
0154  8D  04              BSR     DSPCOM          ;CHAR =)-MAP TO (A) WITH 1 DOTS ABOVE
0156  8D  02              BSR     DSPCOM          ;CHAR =RIGHT BRACE-MAP TO (A) WITH 1 DOT ABOVE
0158  20  1F              BRA     CKSPCE          ;DONE -CHECK SPACE
                                ;
                                ;SUBROUTINE TO COMPARE ACC-A WITH THE FIRST BYTE OF NEXT ROW IN
                                ;       NON-US FONT TABLE.  IF THERE IS A MATCH, A RETURN IS FAKED
                                ;       WITH 2 INS'S AND BRANCH TO CHAR SCANNER.
                                ;
0159  BD  0000G     DSPCOM: JSR     A6X             ;INC POINTER 6 TIMES
015D  A1  00                CMP A   0,X             ;COMPARE ACC-A WITH 1ST BYTE IN ROW
015F  27  01                BEQ     1$
0161  39                    RTS                     ;NOT EQUAL, RETURN
0162  31            1$:    INS                     ;FAKE THE RETURN
0163  31                    INS
0164  08                    INX                     ;INC POINTER TO FIRST DOT PATTERN
0165  20  6D        CHSCB:  BRA     CHSCAN          ;GO TO CHAR SCANNER
                                ;
                                ;REST OF GERMAN
```

```
                                ;
0167  CE  04A9'       DSPSK1: LDX   GERMAN-6,I    ;GERMAN FONT -6 STARTING POINTER
016A  8D  EE                  BSR   DSPCOM        ;CHAR =)-MAP TO (U) WITH 2 DOTS ABOVE
016C  8D  EC                  BSR   DSPCOM        ;CHAR =RIGHT BRACE-MAP TO (U) WITH 2 DOTS ABOVE
016E  20  09                  BRA   CKSPCE        ;DONE-CHECK FOR SPACE
                                ;
```

```
                                        ;REST OF SPANISH
                                        ;
0170  CE 0485'          DSPSK2: LDX     SPANSH-6,I      ;SPANISH POINTER FOR TABLE
0173  8D E5                     BSR     DSPCOM          ;CHAR = ( ?
0175  8D E3                     BSR     DSPCOM          ;CHAR = ¿ ?
0177  8D E1                     BSR     DSPCOM          ;CHAR =) ?
                                        ;
                                        ;
0179  81 1F             CKSPCE: CMP A   37,I
017B  22 0D                     BHI     OFFSET
017D  81 0D                     CMP A   15,I            ;CARRIAGE RETURN?
017F  27 03                     BEQ     1$
0181  7E 0037'                  JMP     NXTTST
0184  7E 0023'          1$:     JMP     CR
0187  7E 026C'          SPJMP:  JMP     SPACE
018A  97 00G            OFFSET: STA A   TMP1,D          ;OFFSET CALC ACC-40 TIMES 5
018C  96 00G                    LDA A   BLINK,D         ; IF WRITE-THRU THEN CAN IGNORE
018E  2B 15                     BMI     2$
0190  96 00G                    LDA A   TMPLX,D
0192  D6 00G                    LDA B   TMPHX,D         ; TEST FOR X AXIS OVERFLOW
0194  C1 03                     CMP B   3,I
0196  2B 0D                     BMI     2$
0198  81 0050G                  CMP A   LINELN          ; TEST FOR LINE LENGTH
019B  25 08                     BCS     2$              ;CAN'T DO YET
019D  BD 010C'                  JSR     CRLF
01A0  96 00G                    LDA A   FULL,D          ; WAS THAT MISSERABLE PAGE FULL?
01A2  27 01                     BEQ     2$
01A4  39                        RTS
01A5  96 00G            2$:     LDA A   TMP1,D          ; GET CHARACTER BACK
01A7  81 20                     CMP A   40,I            ; IF SPACE JUST GO DO IT
01A9  27 DC                     BEQ     SPJMP
01AB  5F                        CLR B
01AC  48                        ASL A
01AD  59                        ROL B
01AE  48                        ASL A
01AF  59                        ROL B
01B0  98 00G                    ADD A   TMP1,D
01B2  C9 00                     ADC B   0,I
01B4  BB 0467'                  ADD A   TABLAD+1
01B7  97 00G                    STA A   TMP3,D
01B9  F9 0466'                  ADC B   TABLAD          ;POSSIBLE NEW X HIGH
01BC  D7 00G                    STA B   TMP2,D          ;X HIGH
01BE  B6 0002G                  LDA A   CHRVCT+2        ; TEST FOR OPTIONAL FONT ROM
01C1  27 0E                     BEQ     1$
01C3  D6 00G                    LDA B   BANK,D
01C5  37                        PSH B
01C6  BD 0000G                  JSR     SETBNK
01C9  FE 0000G                  LDX     CHRVCT
01CC  AD 00                     JSR     0,X
01CE  32                        PUL A
01CF  BD 0000G                  JSR     SETBNK
01D2  DE 00G            1$:     LDX     TMP2,D
01D4  C6 05             CHSCAN: LDA B   5,I
01D6  D7 00G                    STA B   COLCT,D
01D8  F6 0000G                  LDA B   PIAHX
01DB  C4 7F                     AND B   177,I
01DD  D7 00G                    STA B   DOTON,D
```

```
                01DF  7F  0000G           CLR     PIAHY
                01E2  A6  00        NEXT: LDA A   0,X         ;GET COLUMN OF DOTS
                01E4  26  14              BNE     SSCAN       ;SCAN IF SOME DOTS ON
                01E6  F6  0000G           LDA B   PIAHY
                01E9  CB  20              ADD B   #0.1
                01EB  F7  0000G           STA B   PIAHY
                01EE  08                  INX
                01EF  7A  0000G           DEC     COLCT       ;DECREMENT COLUMN COUNT
                01F2  26  EE              BNE     NEXT        ;GET NEXT COLUMN OF DOTS
                01F4  7E  0267'           JMP     FINISH      ;DONE WITH CHARACTER DOTS
```

```
                                        ;SCAN ONE COLUMN OF DOTS
                                        ;
01F7  F7  0000G     GOBABE: STA B  PIAHY
01FA  48            SSCAN:  ASL A
01FB  24  15                BCC    22$
01FD  D6  00G               LDA B  DOTON,D
01FF  F7  0000G             STA B  PIAHX
0202  7D  0000G             TST    BLINK
0205  2B  03                BMI    21$          ;IF TRUE WRITE THRU IS NEEDED-DO IT
0207  7D  0000G             TST    PIAHX
020A  7D  0000G     21$:    TST    PIAHY
020D  CA  80                ORA B  200,I
020F  F7  0000G             STA B  PIAHX
0212  F6  0000G     22$:    LDA B  PIAHY
0215  C8  04                ADD B  4,I
0217  C5  1C                BIT B  34,I
0219  26  DC                BNE    GOBABE
021B  7A  0000G     60$:    DEC    COLCT
021E  27  47                BEQ    FINISH
0220  08                   INX
0221  F6  0000G             LDA B  PIAHY
0224  CB  20                ADD B  40,I
0226  F7  0000G             STA B  PIAHY
0229  A6  00                LDA A  0,X
0228  26  06                BNE    23$
022D  7E  0218'             JMP    60$
0230  F7  0000G     26$:    STA B  PIAHY
0233  44            23$:    LSR A
0234  24  15                BCC    25$
0236  D6  00G               LDA B  DOTON,D
0238  F7  0000G             STA B  PIAHX
023B  7D  0000G             TST    BLINK
023E  2B  03                BMI    24$
0240  7D  0000G             TST    PIAHX
0243  7D  0000G     24$:    TST    PIAHY
0246  CA  80                ORA B  200,I
0248  F7  0000G             STA B  PIAHX
024B  F6  0000G     25$:    LDA B  PIAHY
024E  C5  1C                BIT B  34,I
0250  27  04                BEQ    NEXT1
0252  C0  04                SUB B  4,I
0254  20  DA                BRA    26$
0256  08            NEXT1:  INX
0257  F6  0000G             LDA B  PIAHY
025A  CB  20                ADD B  40,I
025C  F7  0000G             STA B  PIAHY
025F  7A  0000G             DEC    COLCT
0262  27  03                BEQ    FINISH
0264  7E  01E2'             JMP    NEXT
```

```
    1  0267  96    00G      FINISH: LDA A   TMPHY,D
    2  0269  B7  0000G               STA A   PIAHY
    3  026C  D6    00G      SPACE:  LDA B   BLINK,D        ;INHIBIT SPACE IF WRITETHRU-BLINK ON
    4  026E  2B    10               BMI     SPEXT
    5  0270  D6    00G              LDA B   TMPLX,D
    6  0272  4F                     CLR A
    7  0273  CB    0E               ADD B   14.,I
    8  0275  99    00G              ADC A   TMPHX,D
    9  0277  97    00G      UPDATX: STA A   TMPHX,D
   10  0279  8A    FC               ORA A   374,I
   11  027B  D7    00G              STA B   TMPLX,D
   12  027D  BD  0504'              JSR     SETUP
   13  0280  39         SPEXT:  RTS
```

```
                                        .SBTTL  TABLE   TABLE FOR US ASCII CHARACTER FONT
        0281  00  00  5F  TABLE:        .BYTE   0,0,137,0,0             ; !
  0284  00    00
        0286  00  07  00                .BYTE   0,7,0,7,0               ; "
  0289  07    00
        028B  14  7F  14                .BYTE   24,177,24,177,24        ; #
  028E  7F    14
        0290  24  2A  7F                .BYTE   44,52,177,52,22         ; $
  0293  2A    12
        0295  23  13  08                .BYTE   43,23,10,144,142        ;
  0298  64    62
        029A  36  49  56                .BYTE   66,111,126,40,120       ; &
  029D  20    50
        029F  00  04  02                .BYTE   0,4,2,1,0               ; '
  02A2  01    00
        02A4  00  1C  22                .BYTE   0,34,42,101,0           ; (
  02A7  41    00
        02A9  00  41  22                .BYTE   0,101,42,34,0           ; )
  02AC  1C    00
        02AE  22  14  7F                .BYTE   42,24,177,24,42         ; *
  02B1  14    22
        02B3  08  08  3E                .BYTE   10,10,76,10,10          ; +
  02B6  08    08
        02B8  00  80  70                .BYTE   0,260,160,0,0           ; ,
  02BB  00    00
        02BD  08  08  08                .BYTE   10,10,10,10,10          ; -
  02C0  08    08
        02C2  00  60  60                .BYTE   0,140,140,0,0           ; .
  02C5  00    00
        02C7  20  10  08                .BYTE   40,20,10,4,2            ; /
  02CA  04    02
        02CC  3E  51  49                .BYTE   76,121,111,105,76       ; 0
  02CF  45    3E
        02D1  00  42  7F                .BYTE   0,102,177,100,0         ; 1
  02D4  40    00
        02D6  62  51  49                .BYTE   142,121,111,111,106     ; 2
  02D9  49    46
        02DB  21  41  49                .BYTE   41,101,111,115,63       ; 3
  02DE  4D    33
        02E0  18  14  12                .BYTE   30,24,22,177,20         ; 4
  02E3  7F    10
        02E5  27  45  45                .BYTE   47,105,105,105,71       ; 5
  02E8  45    39
        02EA  3C  4A  49                .BYTE   74,112,111,111,61       ; 6
  02ED  49    31
        02EF  01  71  09                .BYTE   1,161,11,5,3            ;
  02F2  05    03
        02F4  36  49  49                .BYTE   66,111,111,111,66       ; 8
  02F7  49    36
        02F9  46  49  49                .BYTE   106,111,111,51,36       ; 9
  02FC  29    1E
        02FE  00  66  66                .BYTE   0,146,146,0,0           ; COLON
  0301  00    00
        0303  00  B6  76                .BYTE   0,266,166,0,0           ; SEMICOLON
  0306  00    00
        0308  08  14  22                .BYTE   10,24,42,101,0          ; <
  030B  41    00
```

```
30                0300  14  14  14      .BYTE   24,24,24,24,24          ; =
                  0310  14  14
31                0312  0G  41  22      .BYTE   0,101,42,24,10          ; >
                  0315  14  08
32                0317  02  01  59      .BYTE   2,1,131,5,2             ; ?
                  031A  05  02
33                031C  3E  41  50      .BYTE   76,101,135,131,16       ;
                  031E  59  0E
34                0321  7C  12  11      .BYTE   174,22,21,22,174        ; A
                  0324  12  7C
35                0326  2F  49  49      .BYTE   177,111,111,111,66      ; B
                  0329  49  36
36                032B  3E  41  41      .BYTE   76,101,101,101,42       ; C
                  032E  41  22
37                0330  7F  41  41      .BYTE   177,101,101,42,34       ; D
                  0333  22  1C
38                0335  7F  49  49      .BYTE   177,111,111,111,101     ; E
                  0338  49  41
39                033A  7F  09  09      .BYTE   177,11,11,11,1          ; F
                  033D  09  01
40                033F  3E  41  41      .BYTE   76,101,101,121,161      ; G
                  0342  51  71
41                0344  7F  08  08      .BYTE   177,10,10,10,177        ; H
                  0347  08  7F
42                0349  00  41  7F      .BYTE   0,101,177,101,0         ; I
                  034C  41  00
43                034E  20  40  40      .BYTE   40,100,100,100,77       ; J
                  0351  40  3F
44                0353  7F  08  14      .BYTE   177,10,24,42,101        ; K
                  0356  22  41
45                0358  7F  40  40      .BYTE   177,100,100,100,100     ; L
                  035B  40  40
46                035D  7F  02  0C      .BYTE   177,2,14,2,177          ; M
                  0360  02  7F
47                0362  7F  04  08      .BYTE   177,4,10,20,177         ; N
                  0365  10  7F
48                0367  3E  41  41      .BYTE   76,101,101,101,76       ; O
                  036A  41  3E
49                036C  7F  09  09      .BYTE   177 11,11,11,6          ; P
                  036F  09  06
50                0371  3E  41  51      .BYTE   76,101,121,41,136       ; Q
                  0374  21  5E
51                0376  7F  09  19      .BYTE   177,11,31,51,106        ; R
                  0379  29  46
52                037B  26  49  49      .BYTE   46,111,111,111,62       ; S
                  037E  49  32
53                0380  01  01  7F      .BYTE   1,1,177,1,1             ; T
                  0383  01  01
54                0385  3F  40  40      .BYTE   77,100,100,100,77       ; U
                  0388  40  3F
55                038A  1F  20  40      .BYTE   37,40,100,40,37         ; V
                  038D  20  1F
56                038F  7F  20  18      .BYTE   177,40,30,40,177        ; W
                  0392  20  7F
57                0394  63  14  08      .BYTE   143,24,10,24,143        ; X
                  0397  14  63
58                0399  03  04  78      .BYTE   3,4,170,4,3             ; Y
```

```
                039C    04      03
59              039E    61      61      49      .BYTE   141,121,111,105,103     ; Z
                03A1    45      43
60              03A3    7F      7F      41      .BYTE   177,177,101,101,101     ; [
                03A6    41      41
61              03A8    02      04      08      .BYTE   2,4,10,20,40            ;
                03AB    10      20
62              03AD    41      41      41      .BYTE   101,101,101,177,177     ; ]
                03B0    7F      7F
63              03B2    04      02      7F      .BYTE   4,2,177,2,4             ; UP-ARROW
                03B5    02      04
64              03B7    80      80      80      .BYTE   200,200,200,200,200     ; UNDERLINE
                03BA    80      80
65              03BC    00      01      02      .BYTE   0,1,2,4,0               ; REVERSE APOSTROPHE
                03BF    04      00
66              03C1    38      44      44      .BYTE   70,104,104,174,100      ; P
                03C4    7C      40
67              03C6    7F      28      44      .BYTE   177,50,104,104,70       ; B
                03C9    44      38
68              03CB    38      44      44      .BYTE   70,104,104,104,0        ; C
                03CE    44      60
69              03D0    38      44      44      .BYTE   70,104,104,50,177       ; D
                03D3    28      7E
70              03D5    38      54      54      .BYTE   70,124,124,124,114      ; E
                03D8    54      4C
71              03DA    00      08      7E      .BYTE   0,10,176,11,11          ; F
                03DD    09      09
72              03DF    98      A4      A4      .BYTE   230,244,244,230,174     ; G
                03E2    98      7C
73              03E4    7F      04      04      .BYTE   177,4,4,4,170           ; H
                03E7    04      78
74              03E9    00      44      7D      .BYTE   0,104,175,100,0         ; I
                03EC    40      00
75              03EE    40      80      80      .BYTE   100,200,200,175,0       ; J
                03F1    7D      00
76              03F3    7F      10      10      .BYTE   177,20,20,50,104        ; K
                03F6    28      44
77              03F8    00      01      7F      .BYTE   0,1,177,0,0             ; L
                03FB    00      00
78              03FD    7C      04      18      .BYTE   174,4,30,4,174          ; M
                0400    04      7C
79              0402    04      7C      04      .BYTE   4,174,4,4,170           ; N
                0405    04      78
80              0407    38      44      44      .BYTE   70,104,104,104,70       ; O
                040A    44      38
81              040C    FC      28      44      .BYTE   374,50,104,104,70       ; P
                040F    44      38
82              0411    38      44      44      .BYTE   70,104,104,50,374       ; Q
                0414    28      FC
83              0416    7C      08      04      .BYTE   174,10,4,4,10           ; R
                0419    04      08
84              0418    48      54      54      .BYTE   110,124,124,124,44      ; S
                041E    54      24
85              0420    00      04      3F      .BYTE   0,4,77,104,104          ; T
                0423    44      44
86              0425    3C      40      40      .BYTE   74,100,100,174,100      ; U
                0428    7C      40
```

```
 87         042A  0C    30   40         .BYTE  14,60,100,60,14        ; V
      042D    30  0C
 88         042F  3C    40   30         .BYTE  74,100,60,100,74       ; W
      0432    40  3C
 89         0434  44    28   10         .BYTE  104,50,20,50,104       ; X
      0437    28  44
 90         0439  9C    A0   A0         .BYTE  234,240,240,240,174    ; Y
      043C    A0  7C
 91         043E  44    64   54         .BYTE  104,144,124,114,104    ; Z
      0441    4C  44
 92         0443  08    08   36         .BYTE  10,10,66,101,101       ; LEFT BRACE
      0446    41  41
 93         0448  00    00   7F         .BYTE  0,0,177,0,0            ; VERTICAL BAR
      044B    00  00
 94         044D  41    41   36         .BYTE  101,101,66,10,10       ; RIGHT BRACE
      0450    08  08
 95         0452  08    04   08         .BYTE  10,4,10,20,10          ; WAVE
      0455    10  08
 96         0457  20    40   FE         .BYTE  40,100,376,100,40      ; DOWN-ARROW (DEL)
      045A    40  20
 97         045C  FF    FF   FF  BLOT:  .BYTE  377,377,377,377,377    ; BLOT AND CURSOR
      045F    FF  FF
 98         0461  F8    F0   E0         .BYTE  320,360,340,320,210    ; PNTR TO GRAPHICS POINT.
      0464    00  88
 99         0466  01DC'          TABLAD: .WORD  TABLE-<41*5>
100                                      ;
```

```
                                        .SBTTL  NONUS   NON-US ASCII CHARACTER FONT
                                ;FONT TABLE FOR THE NON-US CHARACTER FONTS
                                ;
0468  4A  55  55  NONUS:  .BYTE     112,125,125,51,0     ;ALTERNATE FOR  - PARAGRAPH SYMBOL
046B  29  00
0460  7B  08  1C          .BYTE   H7B,10,34,52,10,10     ;ALTERNATE FOR LEFT BRACE - LEFT ARROW
0470  2A  08  08
0473  7C  F8  F0          .BYTE   H7C,37C,36C,34O,32O,21O ;ALTERNATE FOR VERTICAL BAR -PTR TO GRAPHIC PO
                                                                                        INT
0476  E0  00  88
0479  7D  08  08          .BYTE   H7D,10,10,52,34,10     ;ALTERNATE FOR RIGHT BRACE - RIGHT ARROW
047C  2A  1C  08
047F  23  48  7E  GNFONT: .BYTE   H23,110,176,111,111,102;ALTERNATE FOR #-STERLING POUND
0482  49  49  42
0485  24  5A  24          .BYTE   H24,132,44,44,44,132   ;ALTERNATE FOR $-SUN SYMBOL
0488  24  24  5A
0488  58  38  46          .BYTE   H58,70,106,104,176,100 ;ALTERNATE FOR (-A WITHE 2 DOTS
048E  44  7E  40
0491  78  78  25          .BYTE   H78,170,45,42,45,170   ;ALTERNATE FOR LEFT BRACE-A
0494  22  25  78
                                                        ;WITH 2 DOTS ABOVE
0497  5C  38  46          .BYTE   H5C,70,106,104,106,70  ;ALTERNATE FOR  -O WITH 2 DOTS
049A  44  46  38
049D  7C  3C  43          .BYTE   H7C,74,103,102,103,74  ;ALTERNATE FOR VERT BAR-O WITH 2 DOTS
04A0  42  43  3C
04A3  5D  38  44          .BYTE   H5D,70,104,105,174,100 ;ALTER. FOR A WITH 1 DOT ABOVE
04A6  45  7C  40
04A9  7D  78  14          .BYTE   H7D,170,24,23,24,170   ;ALTER. FOR A WITH 1 DOT ABOVE
04AC  13  14  78
04AF  5D  3A  40  GERMAN: .BYTE   H5D,72,100,100,100,72  ;ALTER. FOR )-U WITH 2 DOTS ABOVE
04B2  40  40  3A
04B5  7D  3C  40          .BYTE   H7D,75,100,100,100,75  ;ALTER. FOR RT BRACE-U 2 DOTS
04B8  40  40  3D
04BB  58  00  00  SPANSH: .BYTE   H58,0,0,175,0,0        ;ALTERNATE FOR ( - UPSIDEDOWN !
04BE  7D  00  00
04C1  5C  7D  09          .BYTE   H5C,175,11,21,41,175   ;ALTERNATE FOR BACKSLASH - N WITH A LINE ABOVE
04C4  11  21  7D
04C7  50  30  48          .BYTE   H50,60,110,105,100,40  ;ALTERNATE FOR ) - UPSIDEDOWN ?
04CA  45  40  20
                                ;
```

```
                                        .SBTTL  VECTOR GENERATION
                                ;THESE ROUTINES PUT VECTORS ON THE SCREEN
                                ;DATA WILL BE FROM:
                                ;TMP1-  1 BYTE  FUNCTION (0-DRAW,1-MOVE,2-WRITE THRU)
                                ;TMPHY- 2 BYTES Y DATA (REDUCED TO 10 BITS)
                                ;TMPHX- 2 BYTES X DATA (REDUCED TO 10 BITS)
                                ;
04CD  86  3C          VECTOR:   LDA A   #3C,I  ; DISABLE DOT ONE-SHOTS
04CF  B7  0001G                 STA A   PIAHX+1
04D2  8D  0000'                 JSR     DRBUSY          ; WAIT FOR PRESENT WORK TO FINISH
04D5  B6  0000G                 LDA A   PIAHX           ; MAKE SURE VECTORS ARE OFF!
04D8  84  7F                    AND A   #7F,I
04DA  B7  0000G                 STA A   PIAHX
04DD  D6  00G                   LDA B   TMP1,D          ; TEST VECTOR TYPE 0= DRAW OR WRITE-THRU
                                ;                               1= MOVE
04DF  26  0C                    BNE     MOVE
04E1  85  40                    BIT A   100,I           ; TEST IF VECTOR FILTER ALREADY ON
04E3  27  08                    BEQ     DRAW            ; IF ON THEN READY TO DRAW
04E5  84  3F                    AND A   #3F,I   ; TURN ON FILTER
04E7  B7  0000G                 STA A   PIAHX
04EA  BD  0032'                 JSR     WAIT.           ; NOW WAIT 3.3 MS FOR SETTEL
04ED                    DRAW:
04ED  96  00G           MOVE:   LDA A   TMPHY,D         ; GET Y DATA
04EF  B7  0000G                 STA A   PIAHY
04F2  96  00G                   LDA A   TMPLY,D
04F4  B7  0000G                 STA A   PIALY
04F7  96  00G                   LDA A   TMPHX,D         ; GET X DATA
04F9  D6  00G                   LDA B   TMPLX,D
04FB  7D  0000G                 TST     TMP1            ; DOING A DRAW?
04FE  27  02                    BEQ     2$
0500  8A  40                    ORA A   100,I           ; LEAVE VECTOR FILTER OFF IF MOVE
0502  8A  8C            2$:     ORA A   274,I           ; GET REST OF THE BITS
0504  B7  0000G         SETUP:  STA A   PIAHX
0507  F7  0000G                 STA B   PIALX
050A  39                        RTS
```

```
                                        .SBTTL  CRTINT--INITIALIZE AND RESET CRT
                                ;THESE ROUTINES RESET THE CRT PIA'S AND RESET OLD STATUS IF THE DISPLAY
                                ;WAS DISABLED BY MAGTAPE.
                                        .GLOBL  CRTRST
                                        .GLOBL  DSPSTT          ; DISPLAY STATUS BYTE
                                        .GLOBL  ENABLE          ; ENABLE INTERRUPTS
                        0080    DSPDIS  =  H80                  ; CRT DISABLE FLAG
                        0040    HOMFLG  =  H40                  ; SET IF HOME WAS REQUESTED
                        0020    PAGFLG  =  H20                  ; SET IF PAGE WAS REQUESTED
                        0010    RWNDFG  =  H10                  ; SET IF REWIND REQUESTED
                        0008    CPYFLG  =  H08                  ; SET IF COPY REQUESTED
                                        .GLOBL  CPYFLG
                                        .GLOBL  RWNDFG
                                .
0508    7F      0001G   CRTRST: CLR     PIAHY+1 ; ACCESS DDR
050E    7F      0001G           CLR     PIALY+1
0511    86      FF              LDA A   #0FF.I  ; DATA DIRECTION
0513    B7      0000G           STA A   PIAHY
0516    B7      0000G           STA A   PIALY
0519    86      2C              LDA A   #2C.I   ; CONTROL REG
051B    B7      0001G           STA A   PIAHY+1
051E    B7      0001G           STA A   PIALY+1
0521    96      00G             LDA A   TMPHY,D         ; RESTORE OLD STATUS
0523    B7      0000G           STA A   PIAHY
0526    96      00G             LDA A   TMPLY,D
0528    B7      0000G           STA A   PIALY
052B    96      00G             LDA A   TMPHX,D
052D    8A      FC              ORA A   374,I           ; MAKE IT MOVE THERE
052F    D6      00G             LDA B   TMPLX,D
0531    8D      D1              BSR     SETUP
0533    BD      0578'   1$:     JSR     DSPENL          ; ENABLE DISPLAY
0536    BD      0000G           JSR     ENABLE
0539    39                      RTS                     ;RETURN
```

```
                                        .SBTTL  PCHAR---PRINT CHAR. ON DVST
                                ;THIS ROUTINE IS HAND SHAKE DRIVEN TO ALLOW
                                ;FOR THE SHARED MAGTAPE PIA AND TO RUN THE SYSTEM WITH THE
                                ;INTERUPTS ENABLED EVEN THOUGH PAGE & HOME ARE INTERUPT DRIVEN
053A  D6  000G        PCHAR:  LDA B   DSPSTT,D        ; GET STATUS
053C  2A  0F                  BPL     PCHAR.          ; GO OUTPUT CHAR.
053E  81  0C                  CMP A    HOC,I  ; IF DISABLED HONOR PAGE
0540  26  02                  BNE     1$
0542  CA  20                  ORA B   PAGFLG,I        ; SET PAGE FLAG
0544  81  1E          1$:     CMP H    H1E,I          ; HONOR HOME
0546  26  02                  BNE     2$
0548  CA  40                  ORA B   HOMFLG,I        ; SET HOME FLAG
054A  D7  000G        2$:     STA B   DSPSTT,D        ; SAVE STATUS
054C  39              PCHARE: RTS
054D  CA  80          PCHAR.: ORA B   DSPDIS,I        ; DISPLAY DISABLE
054F  D7  000G                STA B   DSPSTT,D
0551  BD  0000'               JSR     DRBUSY          ; WAIT FOR SCREEN
0554  F6  0000G               LDA B   PIAHX           ; PUT FILTERS ETC. OK
0557  CA  FC                  ORA B   374,I
0559  F7  0000G               STA B   PIAHX
055C  C6  2C                  LDA B    H2C,I          ; MAKE SURE CAN DO CHARACTERS
055E  F7  0001G               STA B   PIAHX+1
0561  D6  000G                LDA B   BLINK,D         ; TURN ON SCREEN IF NOT WRITE-THRU
0563  26  10                  BNE     4$
0565  F6  0001G               LDA B   PIALX+1
0568  2A  08                  BPL     4$              ; TEST IF IN HOLD
056A  F6  0000G               LDA B   PIALX
056D  F7  0000G               STA B   PIALX
0570  36                      PSH A
0571  BD  0032'               JSR     WAIT.
0574  32                      PUL A
0575  BD  0104'          4$:  JSR     PCHR.1          ; GO DO THE CHARACTER

                        ;       I'M CHANGING THE SET/CLI TO TPA/TAP

0578  D6  000G        DSPEN.: LDA B   DSPSTT,D        ; TEST FOR PENDING PAGES
057A  C5  08                  BIT B   CPYFLG,I        ; NEED TO HARDCOPY?
057C  27  03                  BEQ     A.1C
057E  BD  0000G               JSR     DSPCP2
0581  D6  000G        A.1C:   LDA B   DSPSTT,D
0583  C5  20                  BIT B   PAGFLG,I        ; PAGE?
0585  27  05                  BEQ     1$
0587  BD  003A'               JSR     PAGE            ; PAGE THE DISPLAY
058A  20  07                  BRA     2$
058C  C5  40          1$:     BIT B   HOMFLG,I        ; MOVE CURSOR TO HOME
058E  27  03                  BEQ     2$
0590  BD  0014'               JSR     HOME
0593  D6  000G        2$:     LDA B   DSPSTT,D        ; TEST FOR REWIND
0595  7F  0000G               CLR     DSPSTT
0598  C5  10                  BIT B   RWNDFG,I
059A  27  80                  BEQ     PCHARE
059C  BD  0000G               JSR     MTPRWD
                              .GLOBL  MTPRWD
059F  7E  0508'               JMP     CRTRST
              0001'           .END
```

SYMBOL TABLE

```
ABRFLG=  0040        AFAIL =  0030        AMAT  =  0010        ARRAY =  0020        ASTR  =  0020
ATLOD =  0008        ATSNTG=  XXXXXX G    ATVLD =  0004        A.END =  XXXXXX G    A.MAX =  XXXXXX G
A.PRIM=  XXXXXX G    A.PTR =  XXXXXX G    A.SEC =  XXXXXX G    A.STAT=  XXXXXX G    A.STRT=  XXXXXX G
A.1C     0581R       A6X   =  XXXXXX G    BAKSTG=  XXXXXX G    BANK  =  XXXXXX G    BELCAL   0073RG
BFRSIT=  0020        BLINK =  XXXXXX G    BLOT     045CR       BLOIIT   0115R       BMAT  =  0004
BRKCNT=  XXXXXX G    BSTR  =  0008        BUSACI=  0010        CDOPTR=  XXXXXX G    CDSPTR=  XXXXXX G
CHAR  =  XXXXXX G    CHRCNT=  XXXXXX G    CHRVCT=  XXXXXX G    CHSCAN   01D4RG      CHSCB    0165R
CKFONT   0118R       CKSPCE   0179R       CKTAB2   00E8R       CKTAB4   00FAR       CLPTR =  XXXXXX G
CMAT  =  0001        COLCNT=  XXXXXX G    COLCT =  XXXXXX G    CPYFLG=  0008 G      CR       0023R
CRDC3 =  0002        CREOF =  0008        CREOI =  0004        CREOT =  0020        CRETX =  0010
CRLF     010CR       CRNORM=  0001        CRSTAT=  XXXXXX G    CRTDRV=  0000RG      CRIRST   0508RG
CRVLD =  0080        CSTR  =  0002        CTKN  =  XXXXXX G    CURSOR=  XXXXXX G    DATDEV=  0022
DELY1S=  XXXXXX G    DIMFLG=  0004        DIRCT =  0080        DISCNT=  XXXXXX G    DISSRQ=  0008
DL    =  XXXXXX G    DOTON =  XXXXXX G    DP    =  XXXXXX G    DRAW     04E0R       DRBUSY   0000RG
DREXTA=  XXXXXX G    DREXTB=  XXXXXX G    DSHOME   0014RG      DSPAGE   000ARG      DSPCOM   015AR
DSPCP2=  XXXXXX G    DSPDEV=  0020        DSPDIS=  0080        DSPENL   0578R       DSPSK1   0167R
DSPSK2   0170R       DSPSIT=  XXXXXX G    DT    =  XXXXXX G    EDTBFR=  XXXXXX G    ENABLE=  XXXXXX G
ENDKEY=  0040        ETYP=    0038        EOLTG =  XXXXXX G    EOSTG =  XXXXXX G    ERATSN=  XXXXXX G
ERDOMN=  XXXXXX G    ERFLOM =  XXXXXX G   ERFBFR=  XXXXXX G    ERFILE=  XXXXXX G    ERIOE =  XXXXXX G
ERNIOD=  XXXXXX G    ERNSEP=  XXXXXX G    ERRCD =  XXXXXX G    ERTERM=  XXXXXX G    ERUNDE=  XXXXXX G
ESTG  =  XXXXXX G    EXTFLG=  0080        FILDEV=  0000        FINISH   0267R       FIX1A =  0003
FIX1B =  0004        FMTVLD=  0008        FNFLG =  0010        FONT  =  XXXXXX G    FORTG =  XXXXXX G
FULL  =  XXXXXX G    GERMAN   04AFR       GLBFLG=  XXXXXX G    GNFONT   047FR       GOBACE   01F7R
GOSTG =  XXXXXX G    GOUPX    00F1R       HOME     0014R       HOMFLG=  004C        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 =  001E        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    LF       003BR       LINELN=  XXXXXX G
LISTTG=  XXXXXX G    LNNOTG=  XXXXXX G    LOCTG =  XXXXXX G    LSP   =  XXXXXX G    LSTEMT=  0002
MOVE     04E0R       MTBFR =  XXXXXX G    MTPDEV=  0021        MTPD2 =  0023        MTPRMD=  XXXXXX G
NEXT     01E2R       NEXT1    0256R       NLPTR =  XXXXXX G    NOKEY =  0080        NONUS    0468R
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    NXTTST   0037R       OBJATR=  0002        OBJBCK=  0003        OBJDT =  0005
OBJLEN=  0000        OFFSET   018AR       ONSFLG=  0002        OPRADR=  XXXXXX G    PAETG =  XXXXXX G
PAGE     000AR       PAGFLG=  0020        PARM  =  0008        PCHAR    053ARG      PCHARE   054CR
PCHARZ   0111RG      PCHBR    054DR       PCHR 1   0104R       PGMATR=  0002        PGMBP =  0005
PGMCD =  0009        PGMFP =  0003        PGMLEN=  0000        PGMLNN=  0007        PGMPTR=  XXXXXX G
PGMTG =  XXXXXX G    PIAHX =  XXXXXX G    PIAHY =  XXXXXX G    PIALT =  XXXXXX G    PIALX =  XXXXXX G
PIALY =  XXXXXX G    PLOSTG=  XXXXXX G    PNDEOF=  XXXXXX G    PNDFLG=  XXXXXX G    PNTNTG=  XXXXXX G
PNTSTG=  XXXXXX G    POINT =  XXXXXX G    PPMODE=  XXXXXX G    PRIDEF=  0001        PRTTG =  XXXXXX G
PSCTG =  XXXXXX G    PUPD     0055R       RECLFG=  0004        RFAIL =  00C0        RMAT  =  0040
RPTCTL=  0080        RSTR  =  0080        RTRNTG=  XXXXXX G    RUNFLG=  0080        RUNN  =  0002
RWNDFG=  0010 G      R0    =  XXXXXX G    R1    =  XXXXXX G    R10   =  XXXXXX G    R11   =  XXXXXX G
R12   =  XXXXXX G    R13   =  XXXXXX G    R14   =  XXXXXX G    R15   =  XXXXXX G    R16   =  XXXXXX G
R17   =  XXXXXX G    R18   =  XXXXXX G    R19   =  XXXXXX G    R2    =  XXXXXX G    R20   =  XXXXXX G
R21   =  XXXXXX G    R22   =  XXXXXX G    R23   =  XXXXXX G    R3    =  XXXXXX G    R4    =  XXXXXX G
R5    =  XXXXXX G    R6    =  XXXXXX G    R7    =  XXXXXX G    R8    =  XXXXXX G    R9    =  XXXXXX G
SBP   =  XXXXXX G    SCALER=  0040        SECDEF=  0002        SEMITG=  XXXXXX G    SETBNK=  XXXXXX G
SETUP    0504R       SET2     00EFR       SET3     00F4R       SET4     00FER       SNDIT =  0080
SPACE    026CR       SPANSH   0488R       SPEXT    0280R       SPJMP    0187R       SSCAN    01FAR
STAT37=  XXXXXX G    STPFLG=  0040        STPKEY=  0020        STRING=  0010        SYSERR=  XXXXXX G
TABLAD   0466R       TABLE    0281R       TABPTR=  XXXXXX G    TAGPTR=  XXXXXX G    TCOL  =  XXXXXX G
THOME    0061R       TMPHX =  XXXXXX G    TMPHY =  XXXXXX G    TMPLX =  XXXXXX G    TMPLY =  XXXXXX G
TMP1  =  XXXXXX G    TMP2  =  XXXXXX G    TMP3  =  XXXXXX G    TRCFLG=  0020        TSTBS    0097R
T51DEV=  0025        UNDEF =  0080        UPDATE   0050R       UPDATX   0277R       VALTG =  XXXXXX G
```

```
CRTDRV DISPLAY DRIVER  RT-11 MMAC  VM02-10   14-OCT-76 01:29:57 PAGE 14+
SYMBOL TABLE

VALUNG=  0040        VECTOR   04CDRG       WAIT.    0032RG       XAXIS = xxxxxx G
  ABS.   0000    00
         05A2    01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2171.  WORDS
.SY:CRTDRV/C(DK1:SE1CL1.CRTDKV
```

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)

CROSS REFERENCE TABLE (CREF V01-03.)

ISTBS	3-19	4-3#				
UNDEF	1-81#					
UPDATE	3-3	3-5#	4-31			
UPDATX	5-18	9-9#				
VALTG	1-134#					
VALUND	1-88#					
VECTOR	1-280#	12-8#				
WAIT.	2-41#	2-44#	4-16	12-21	14-30	
XAXIS	1-248#					

CROSS REFERENCE TABLE (CREF V01-03)

SE1 1-3%

DDDDDDDDD EEEEEEEEEE LL EEEEEEEEEE TTTTTTTTTT EEEEEEEEEE LL SSSSSSSS TTTTTTTTTT
DDDDDDDDDD EEEEEEEEEE LL EEEEEEEEEE TTTTTTTTTT EEEEEEEEEE LL SSSSSSSSS TTTTTTTTTT
DD DD EE LL EE TT EE LL SS S TT
DD DD EE LL EE TT EE LL SS TT
DD DD EEEEEE LL EEEEEE TT EEEEEE LL SSSSSSSS TT
DD DD EEEEEE LL EEEEEE TT EEEEEE LL SSSSSSSS TT
DD DD EE LL EE TT EE LL SS TT
DD DD EE LL EE TT EE LL S SS TT
DDDDDDDDD EEEEEEEEEE LLLLLLLLLL EEEEEEEEEE TT EEEEEEEEEE ..... LLLLLLLLLL SSSSSSSSS TT
DDDDDDDDD EEEEEEEEEE LLLLLLLLLL EEEEEEEEEE TT EEEEEEEEEE .... LLLLLLLLLL SSSSSSSS TT

```
                                        .IDENT  /MCDA1A/
                                        .GLOBL  DELETE
                0000'                   DELETE=.
                                        .TITLE  DELETE
                                        .GLOBL  DELET, FIX1, A9X, SYSERR, A5X, GETSMA, GETLAR
                                        .GLOBL  VALTG, PNTNTG, EOLTG, ALLTG, PNTSTG, BACKUP
                                        .GLOBL  R0, R1, R2, R3, R4, R5, R6, R7
                                        .GLOBL  LOCTGR, JMPX,OLTALL,IDLE
                                        .GLOBL  LCLFLG,ERFIXN,ERRCD,PSHRET,RTRN
                                ; EXPECTS ONE OR TWO STATEMENT NUMBERS
                                ;   (THE NUMBER OF THE STATEMENT TO BE DELETED
                                ;    OR THE NUMBERS OF THE STARTING AND ENDING
                                ;    STATEMENTS OF A BLOCK OF STMTS TO BE
                                ;    DELETED)
                                ; OR A LIST OF POINTERS TO THE NT ENTRIES FOR VARIABLES
                                ; TO BE DELETED
                                ; REG USAGE
                                ;   R0 - COMMUNICATIONS TO LOCTGR
                                ;   R1 - TEMP FOR LOCTGR
                                ;   R2 - RETURN ADDR FROM DELETE
                                ;   R3 - START DELETE # (LATER PREDECESSOR #)
                                ;   R4 - STOP DELETE # (LATER SUCCESSOR #)
                                ;   R5 - PTR TO PREDECESSOR
                                ;   R6 - PTR O SUCCESSOR
                                ;   R7 - TEMP
0000    8D    0000G                       JSR     PSHRET
0003    96      00G                       LDA A   LCLFLG,D        ; SET THE EVALUATORS
0005    8A      40                        ORA A    H40,I  ; ABORT BIT
0007    97      00G                       STA A   LCLFLG,D
0009    86      00G                       LDA A   VALTG,I         ; IS STACK (TOP) A
000B    30                                TSX                     ; FPN
000C    A1      00                        CMP A   0,X
000E    27      15                        BEQ     STMTS           ; IF SO DELETE STATEMENTS
0010    86      00G                       LDA A   ALLTG,I
0012    A1      00                        CMP A   0,X
0014    27      71                        BEQ     DELALL
0016    7E    0065'                       JMP     VARS            ; IF NOT DELETE VARIABLES
0019    27      09              DELFIX:   BEQ     1$              ;NO ERRORS
001B    80      00G                       SUB A   ERFIXN,I        ;CHECK FOR "FIX OF A NEG NUMBER"
001D    26      05                        BNE     1$              ;REALLY WAS AN OVERRANGE ERROR
001F    A7      03                        STA A   3,X             ;WE FELL THRU, ACC-A IS ZERO
0021    4C                                INC A                   ;
0022    A7      04                        STA A   4,X             ;CHANGE LOW LINE # TO A ONE
0024    39                      1$:       RTS
0025    30              STMTS:  TSX
0026    8D    0000G                       JSR     FIX1            ; INTEGRIZE STOPPING
                                                                  ; STMT NUMBER
0029    8D      EE                        BSR     DELFIX          ;CK ERROR AND MAYBE FIX UP LINE #
002B    EE      03                        LDX     3,X             ; AND SAVE IN R4
002D    DF      00G                       STX     R4,D
002F    30                                TSX
0030    8D    0000G                       JSR     A9X             ; DELETE THIS ENTRY
0033    35                                TXS                     ; FROM THE STACK
0034    86      00G                       LDA A   VALTG,I         ; IF NEXT ENTRY IS NOT
0036    A1      00                        CMP A   0,X             ; ANOTHER FPN WE ARE DELETING
0038    26      13                        BNE     JSTWON          ; JUST ONE STATEMENT
003A    8D    0000G                       JSR     FIX1            ; IF IT IS A FPN, INTEGERIZE
```

```
                                                        ; IT AND STORE IT IN
0030  8D  0A                  BSR   DELFIX              ;CK ERROR AND MAYBE FIX UP LINE #
003F  7F  0000G               CLR   ERRCD               ;CLEAR ALL ERRORS AND GO, BABE.
0042  EE  03                  LDX   3,X                 ; R3
0044  DF  00G                 STX   R3,D
0046  30                     TSX
0047  BD  0000G               JSR   A9X                 ; THEN DELETE IT
004A  35                     TXS                       ; FROM THE STACK
004B  20  04                  BRA   READY
004D  DE  00G         JSTWON: LDX   R4,D                ; COPY R3 INTO R4 (ENDING STMT #
004F  DF  00G                 STX   R3,D                ;EQUALS STARTING STMT #)
0051  7F  0000G       READY:  CLR   ERRCD
0054  BD  008D'               JSR   DELET               ; "DELET" IS THE ROUTINE WHICH
                                                        ; ACTUALLY DOES THE DELETING OF STMTS
0057  7E  0000G       DONE:   JMP   RTRN                ; RETURN
005A  EE  01          NMERIC: LDX   1,X                 ; SET TO SIMPLE UNDEFINED
005C  86  C0                  LDA A   #0C0,I
005E  A7  04                  STA A   4,X
0060  30                     TSX
0061  BD  0000G               JSR   A9X                 ; CLEAN UP STACK
0064  35                     TXS
0065  86  00G         VARS:   LDA A   PNTNTG,I          ; IF THE STACK (TOP) IS  NT NUMERIC
0067  A1  00                  CMP A   0,X               ; GO TO NMERIC
0069  27  EF                  BEQ     NMERIC
0068  86  00G                 LDA A   PNTSTG,I          ; IF IT IS  NT STRING
006D  A1  00                  CMP A   0,X               ; GO TO STRING
006F  27  09                  BEQ     STRING
0071  86  00G                 LDA A   EOLTG,I           ; IF NOT EITHER THEN IF NOT
0073  A1  00                  CMP A   0,X               ; AN END-OF-LINE WE HAVE A
0075  27  E0                  BEQ     DONE              ; SYSTEM ERROR
                                                        ; IF IT IS AN END-OF-LINE, QUIT
0077  BD  0000G       ARRGH:  JSR   SYSERR
007A  EE  01          STRING: LDX   1,X                 ; SET TO STRING UNDEFINED
007C  86  90                  LDA A   #90,I
007E  A7  04                  STA A   4,X
0080  30                     TSX
0081  BD  0000G               JSR   A5X                 ; CLEAN UP STACK
0084  35                     TXS
0085  20  0E                  BRA   VARS                ; GO LOOK FOR ANOTHER
0087  BD  0000G       DELALL: JSR   DLTALL
008A  7E  0000G               JMP   IDLE
```

```
                                ; THE FOLLOWING ROUTINE EXPECTS TWO LINE
                                ; NUMBERS IN R3 AND R4 (THEY MAY BE THE SAME).
                                ; IT DELETES THE LINES FROM (R3) TO (R4)
                                ; INCLUSIVE
                                        .GLOBL  LNNOTG,PLOSTG,LISTTG,ESTG,ONTBL,PGMPTR,A6X,CLPTR
                                        .GLOBL  NLPTR,RTRNTG,R8,IMXTG,LOCTG,CDOPTR,CDSPTR,PUSHXT,ZX

0080    86      000     DELET:  LDA A   RTRNTG,I        ; TAG RET. ADDR.
008F    36                      PSH A
0090    DE      000             LDX     R3,D            ; FIND PREDECESSOR TO
0092    27      01              BEQ     1$
0094    09                      DEX                     ; STARTING LINE
0095    DF      000     1$:     STX     R8,D
0097    96      01G             LDA A   R8+1,D
0099    36                      PSH A
009A    91      01G             CMP A   R4+1,D
009C    96      000             LDA A   R8,D
009E    36                      PSH A
009F    92      000             SBC A   R4,D
00A1    25      04              BCS     OK
00A3    32                      PUL A
00A4    32                      PUL A
00A5    32                      PUL A
00A6    39                      RTS
00A7    86      000     OK:     LDA A   LNNOTG,I
00A9    36                      PSH A
00AA    BD      0000G           JSR     GETSMA
00AD    30                      TSX
00AE    EE      01              LDX     1,X
00B0    26      09              BNE     1$              ; STORE POINTER TO PREDECESSOR
00B2    30                      TSX
00B3    4F                      CLR A
00B4    A7      04              STA A   4,X
00B6    A7      05              STA A   5,X
00B8    CE      FFFDG           LDX     <PGMPTR-3),I
00BB    DF      000     1$:     STX     R5,D            ; IN R5
00BD    30                      TSX
00BE    EE      04              LDX     4,X             ; STORE PREDECESSOR'S LINE
00C0    DF      000             STX     R3,D            ; # IN R3
00C2    30                      TSX
00C3    BD      0000G           JSR     A6X
00C6    35                      TXS
00C7    DE      000             LDX     R4,D            ; FIND SUCCESSOR TO
00C9    8C      FFFF            CPX     65535.;         ; COMPARE X TO  HOFFFF (MC BUG HERE)
00CC    27      1A              BEQ     5$              ; ENDING LINE #
00CE    08                      INX
00CF    86      000             LDA A   LNNOTG,I
00D1    BD      0000G           JSR     PUSHXT
00D4    BD      0000G           JSR     GETLAR
00D7    30                      TSX
00D8    EE      01              LDX     1,X             ; STORE POINTER TO SUCCESSOR
00DA    DF      000             STX     R6,D            ; IN R6
00DC    30                      TSX
00DD    EE      04              LDX     4,X
00DF    DF      000             STX     R4,D            ; STORE LINE # OF SUCCESSOR
00E1    30                      TSX                     ; IN R4
00E2    BD      0000G           JSR     A6X
```

```
                        TXS
                        BRA     6$
5$:                     LDX     ZX,0
                        STX     R6,0
6$:                     SEI
.BYTE   01,17
                        LDX     R5,0            ; FORWARD POINTER OF PREDECESSOR
                        LDA A   R6,0            ; IS SET TO POINTER TO
                        LDA B   (R6+1),0        ; SUCCESSOR
                        STA A   3,X
                        STA B   4,X
                        LDX     R6,0            ; BACK POINTER OF SUCCESSOR
                        BEQ     2$
                        LDA B   R5+1,0          ; IS SET EQUAL TO THE
                        LDA A   R5,0            ; POINTER TO THE PREDECESSOR
                        BNE     3$
                        CLR B
3$:                     STA A   5,X
                        STA B   6,X
2$:                     CLI
                        LDX     CDSPTR
                        BEQ     CURLIN
                        BSR     DELCHK
                        BRA     CURLIN
                        LDX     ZX,0
                        STX     CDSPTR
                        STX     CDSPTR
CURLIN:                 LDX     CLPTR,0         ; ARE WE DELETING THE CURRENT
                        BEQ     NEXLIN
                        BSR     DELCHK          ; LINE
                        BRA     NEXLIN
                        LDA A   R5,0            ; IF WE ARE SET CLPTR TO
                        STA A   CLPTR,0         ; PREDECESSOR LINE
                        LDA A   (R5+1),0
                        STA A   (CLPTR+1),0
NEXLIN:                 LDX     NLPTR,0
                        BEQ     STACK
                        BSR     DELCHK
                        BRA     STACK
                        LDA A   R6,0
                        STA A   NLPTR,0
                        LDA A   R6+1,0
                        STA A   NLPTR+1,0
                                                ; WANT SCAN TO START AT
STACK:  STS     R0,0            ; THE TOP OF THE STACK
                                ; (ON SUCCESSIVE CALLS CONTINUE ON DOWN
SCAN:   LDA A   PLOSTG,I        ; STACK FROM  POINT WHERE LEFT OFF)
        LDA B   LISTTG,I        ; AND SCAN FOR THE RANGE CONTAINING
        JSR     LOCTGR          ; LINE POINTERS
                                ; IF X IS NULL WE'RE
        BEQ     NEXTJP          ; DONE WITH THIS PHASE
        LDX     2,X             ; LOAD THE POINTER TO THE LINE
        BSR     DELCHK          ; IS THIS LINE BEING DELETED
        BRA     EVLNXT          ; IF NOT CHECK FOR NEED TO
                                ; UPDATE EVAL NLPTR
                                ; IF SO MUST DO SOME UPDATING
        LDA A   1,X
```

```
                                CMP A   PLOSTG,I        ; IF THIS IS A  LITERAL IN OBJ
                                BEQ     LITSTR          ; STRING GO TO LITSTR
                                CMP A   ESTG,I          ; IS THIS AN EVAL. STATUS ENTRY?
                                BNE     1$
                                LDA B   9.,X            ; SET ABORT BIT IN
                                ORA B    H40,I  ; EVAL. STATUS
                                STA B   9.,X
                1$:             LDX     R0,D            ; SET LINE
                                LDA A   R5,D            ; PTR EQUAL TO PTR
                                STA A   2,X             ; TO THE PREDECESSOR
                                LDA A   (R5+1),D        ; LINE
                                STA A   3,X
                                BRA     EVLNXT
                NEXTUP:         BRA     NEXT
                DELCHK:         LDA A   R5,D
                                BNE     1$
                                LDA A   (R5+1),D
                                BEQ     R6MABE
                1$:             LDA A   R6,D
                                BNE     2$
                                LDA A   (R6+1),D
                                BEQ     R5ONLY
                2$:             LDA A   7,X
                                CMP A   R4,D
                                BHI     NO
                                LDA A   8.,X
                                CMP A   (R4+1),D
                                BCC     NO
                R5ONLY:         LDA A   7,X
                                CMP A   R3,D
                                BCS     NO
                                LDA A   8.,X
                                CMP A   (R3+1),D
                                BLS     NO
                                BRA     YES
                R6MABE:         LDA A   R6,D
                                BNE     3$
                                LDA A   (R6+1),D
                                BEQ     NO
                3$:             LDA A   7,X
                                CMP A   R4,D
                                BHI     NO
                                LDA A   8.,X
                                CMP A   (R4+1),D
                                BCC     NO
                YES:            TSX             ,MD'S DIRTY CODE
                                LDA A   1,X
                                ADD A   2,I
                                STA A   1,X
                                BCC     NO
                                INC     0,X
                NO:             LDX     R0,D
                                RTS
                LITSTR:
                                CLR A                   ; FOR  TO LITERAL IN THE
                                STA A   2,X             ; OBJ STRING SET THE LINE
                                STA A   3,X             ; POINTER AND CHAR PTR
```

```
0189  A7  04              STA A  4,X            ; TO NULL
018B  A7  05              STA A  5,X
018D  86  00G    EVLNXT:  LDA A  ESTG,I         ; SEE IF THIS IS A
018F  A1  01              CMP A  1,X            ; EVAL STATUS ENTRY
01C1  26  0E              BNE    AGAIN
01C3  EE  04              LDX    4,X            ; IF SO NEED TO CHECK
01C5  8D  A1              BSR    DELCHK         ; THE NEXT LINE POINTER ALSO.
01C7  20  08              BRA    AGAIN          ; IS THE LINE POINTED TO BY
                                                ; THE NEXT LNE POINTER BEING
                                                ; DELETED?
                                                ; IF SO SET
01C9  96  00G             LDA A  R6,D           ; IT TO SUCCESSOR POINTER
01CB  A7  04              STA A  4,X
01CD  96  01G             LDA A  (R6+1),D
01CF  A7  05              STA A  5,X
01D1  BD  0000G  AGAIN:   JSR    BACKUP
01D4  7E  0138'           JMP    SCAN
01D7             NEXT:                          ; SEARCH THE STACK FOR POINTERS TO THE
01D7  9F  00G             STS    R0,D           ; LINE(S) BEING DELETED AND IF ANY
01D9  86  00G    1$:      LDA A  IMYTG,I        ; ARE FOUND NULL THEM OUT
01DB  BD  0000G           JSR    LOCTG
01DE  27  10              BEQ    TABLES
01E0  EE  02              LDX    2,X
01E2  8D  84              BSR    DELCHK
01E4  20  05              BRA    2$
01E6  5F                  CLR B
01E7  E7  02              STA B  2,X
01E9  E7  03              STA B  3,X
01EB  BD  0000G  2$:      JSR    BACKUP
01EE  20  E9              BRA    1$
01F0  CE  0000G  TABLES:  LDX    ONTBL,I        ; CHECK ONTABL, EOFTABL
01F3  DF  00G             STX    R7,D           ; AND DEFTABL TO SEE IF
01F5  C6  29              LDA B  41.,I          ; WE'VE DELETED ONE OF
01F7  EE  00     LOOP:    LDX    0,X            ; THESE LINES
01F9  BD  0168'           JSR    DELCHK
01FC  20  07              BRA    NOCHG
01FE  DE  00G             LDX    R7,D           ; IF SO ZERO THE ENTRY
0200  4F                  CLR A
0201  A7  00              STA A  0,X
0203  A7  01              STA A  1,X
0205  DE  00G    NOCHG:   LDX    R7,D           ; IF NOT LOOK AT NEXT
0207  08                  INX                   ; TABLE ENTRY IF THERE IS
0208  08                  INX                   ; ONE
]]]]]
0209  DF  00G             STX    R7,D
020B  5A                  DEC B
020C  26  E9              BNE    LOOP           ; THERE ARE 41 TABLE ENTRIES
020E  32                  PUL A                 ; UNTAG RET. ADDR.
020F  39                  RTS
      0001'               END
```

```
DELETE RT-11 MMAC  VM02-10   14-OCT-75 01:30:21 PAGE 2+
SYMBOL TABLE

AGAIN    0101R       ALLTG = XXXXXX G      ARRGH    0077R       A5X   = XXXXXX G      A6X   = XXXXXX G
A9X   = XXXXXX G     BACKUP= XXXXXX G      CDOPTR= XXXXXX G      CDSPTR= XXXXXX G      CLPTR = XXXXXX G
CURLIN   0119R       DELALL   0087R        DELCHK   0168R       DELET    0G8DRG       DELETE=   0000RG
DELFIX   0019R       DLTALL= XXXXXX G      DONE     0057R       EOLTG = XXXXXX G      ERFIXN= XXXXXX G
ERRCD = XXXXXX G     ESTG  = XXXXXX G      EVLNXT   018DR       FIX1  = XXXXXX G      GETLOR= XXXXXX G
GETSMA= XXXXXX G     IDLE  = XXXXXX G      IMXTG = XXXXXX G      JMPX  = XXXXXX G      JSTION    0040R
LCLFLG= XXXXXX G     LISTTG= XXXXXX G      LITSTR   0184R       LNNOTG= XXXXXX G      LOCTG = XXXXXX G
LOCTGR= XXXXXX G     LOOP     01E7R        NEXLIN   0129R       NEXT     01D7R        NEXTJP    0166R
NLPTR = XXXXXX G     NMERIC   005AR        NO       0181R       NOCHG    0205R        OK        00A7R
ONTBL = XXXXXX G     PGMPTR= XXXXXX G      PLOSTG= XXXXXX G      PNTNTG= XXXXXX G      PNTSTG= XXXXXX G
PSHRET= XXXXXX G     PUSHXT= XXXXXX G      READY    0051R       RTRN  = XXXXXX G      RTRNTG= XXXXXX G
R0    = XXXXXX G     R1    = XXXXXX G      R2    = XXXXXX G      R3    = XXXXXX G      R4    = XXXXXX G
R5    = XXXXXX G     R5ONLY   0184R        R6    = XXXXXX G      R6.1ABE  0192R       R7    = XXXXXX G
R8    = XXXXXX G     SCAN     0138R        SCRCK    0139R       STMTS    0U25R        STRING    007AR
SYSERR= XXXXXX G     TABLES   01F0R        VALTG = XXXXXX G      VARS     0065R        YES       01A6R
.X    = XXXXXX G
  ABS.    0000     00
          0210     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3109. WORDS
.SY:DELETE/C:DK1:SELCLL,DELETE
```

CROSS REFERENCE TABLE (CREF V01-03)

1-18

A5X 1-20# 1-109
A6X 2-5# 2-41 2-57
A9X 1-20# 1-62 1-79 1-92
AGAIN 2-175 2-178 2-186#
ALLTG 1-21# 1-49
ARRGH 1-104#
BACKUP 1-21# 2-186 2-199
COPTR 2-6# 2-82*
COSPTR 2-6# 2-77 2-83*
CLPTR 2-5# 2-84 2-89* 2-91*
CURLIN 2-78 2-80 2-84#
DELALL 1-51 1-112#
DELCHK 2-79 2-86 2-94 2-109 2-128# 2-177 2-194 2-205
DELET 1-20# 1-85 2-8#
DELETE 1-12# 1-18#
DELFIX 1-53# 1-63 1-74
DLTALL 1-23# 1-112
DONE 1-82 1-102
EOLTG 1-21# 1-100
EPFIXN 1-24# 1-54
ERRCO 1-24# 1-75* 1-84*
ESTG 2-5# 2-116 2-173
EVLNXT 2-110 2-126 2-173#
FIX1 1-20# 1-61 1-72
GETLAR 1-20# 2-49
GETSMA 1-20# 2-27
IDLE 1-23# 1-113
IMXTG 2-6# 2-190
JMPX 1-23#
JSTHON 1-71 1-82#
LCLFLG 1-24# 1-42 1-44*
LISTTG 2-5# 2-104
LITSTR 2-115 2-167*
LNNOTG 2-5# 2-25 2-47
LOCTG 2-6# 2-191
LOCTGR 1-23# 2-105
LOOP 2-204# 2-216
NEXLIN 2-85 2-87 2-92#
NEXT 2-127 2-188#
NEXTJP 2-107 2-127#
NLPTR 2-6# 2-92 2-97* 2-99*
NMERIC 1-88# 1-96
NO 2-138 2-141 2-144 2-147 2-152 2-155 2-158 2-163 2-165#
NOCHG 2-206 2-211#
OK 2-20 2-25#
ORITBL 2-5# 2-201
PGMPTR 2-5# 2-35
PROSTG 2-5# 2-103 2-114
PNTNTG 1-21# 1-94
PNTSTG 1-21# 1-97
PSHRET 1-24# 1-41
PUSHXT 2-6# 2-48
R0 1-22# 2-101* 2-121 2-165 2-189*
R1 1-22#
R2 1-22#

CROSS REFERENCE TABLE (CREF V01-03.)

R3	1-22#	1-77*	1-83*	2-10	2-39*	2-143	2-146							
R4	1-22#	1-65*	1-8?	2-16	2-19	2-43	2-55*	2-137	2-140	2-154	2-157			
R5	1-22#	2-36*	2-6?	2-70	2-71	2-88	2-90	2-122	2-124	2-128	2-130			
R6ONLY	2-135	2-142#												
R6	1-22#	2-52*	2-61*	2-64	2-65	2-68	2-96	2-98	2-132	2-134	2-149	2-151	2-182	2-184
R6MABE	2-131	2-149#												
R7	1-22#	2-202*	2-207	2-211	2-214*									
R8	2-6#	2-13*	2-14	2-17										
READY	1-81	1-84#												
RTRN	1-24#	1-87												
RTRNTG	2-6#	2-8												
SCAN	2-103#	2-187												
STACK	2-93	2-96	2-101#											
STMTS	1-48	1-60#												
STRING	1-99	1-105#												
SYSERR	1-20#	1-104												
TABLES	2-192	2-201#												
VALTG	1-21#	1-45	1-69											
VARS	1-52	1-94#	1-111											
YES	2-148	2-159#												
ZX	2-6#	2-60	2-81											

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3# 2-62

```
DDDDDDDD   IIIIIIIIII  MM     MM   SSSSSSSS   UU     UU  BBBBBBBB            LL          SSSSSSSS  TTTTTTTTTT
DDDDDDDDD  IIIIIIIIII  MMMM MMMM  SSSSSSSSSS  UU     UU  BBBBBBBBBB          LL          SSSSSSSSSS TTTTTTTTTT
DD     DD     II      MM MMMM MM  SS      S   UU     UU  BB     BB           LL          SS      S     TT
DD     DD     II      MM  MM  MM  SS          UU     UU  BB     BB           LL          SS            TT
DD     DD     II      MM  MM  MM  SSSSSSSSS   UU     UU  BBBBBBBBB           LL          SSSSSSSSS     TT
DD     DD     II      MM      MM   SSSSSSSSS  UU     UU  BBBBBBBBB           LL           SSSSSSSSS    TT
DD     DD     II      MM      MM         SS   UU     UU  BB     BB           LL                SS    TT
DD     DD     II      MM      MM  S      SS   UU     UU  BB     BB           LL          S     SS    TT
DDDDDDDDDD IIIIIIIIII  MM      MM  SSSSSSSSSS  UUUUUUUUUU  BBBBBBBBB  .....  LLLLLLLLLL  SSSSSSSSSS    TT
DDDDDDDD   IIIIIIIIII  MM      MM   SSSSSSSS    UUUUUUUU   BBBBBBBB   ....  LLLLLLLLLL   SSSSSSSS     TT
```

```
                                .TITLE  DIMSUB  DIMENSION/SUBSCRIPTING
                                .GLOBL  DIMSUB
0000'                           DIMSUB=.
                                .IDENT  /JK0016/
                                .GLOBL  RPN             ; RIGHT PARENTHESIS.
                                .GLOBL  TYPARG          ; GET ARGUMENT TYPE/VALUE.
                                .GLOBL  FIX1            ; FP TO INTEGER.
                                .GLOBL  INTMLA          ; INTEGER MULTIPLY ACCUMULATE.
                                .GLOBL  COMPR           ; COMPRESS MEMORY.
                                .GLOBL  A8X             ; INCREMENT X BY 8.
                                .GLOBL  INTMLC          ; INTEGER MULTIPLY, NON-ACCUMULATE.
                                .GLOBL  DIMSTR          ; DIMENSION STRING.
                                .GLOBL  DIM             ; DIMENSION HANDLER.
                                .GLOBL  BYTCAL          ; DIM/SUB CALCULATOR.
                                .GLOBL  DISBLE          ; DISABLE BREAKS.
                                .GLOBL  ENABLE          ; ENABLE BREAKS.
                                .GLOBL  PSHRET          ; PUSH RETURN ONTO PSEUDO STACK.
                                .GLOBL  RTRN            ; RETURN USING PSEUDO STACK.


                                .GLOBL  INT1            ; ROW DIMENSION/SUBSCRIPT.
                                .GLOBL  INT2            ; COLUMN DIMENSION/SUBSCRIPT.
                                .GLOBL  BRKCNT          ; LEFT BRACKET COUNT.
                                .GLOBL  LCLFLG          ; LOCAL FLAG.
0004                            DIMFLG=4                ; DIMENSION FLAG.
                                .GLOBL  R0              ; WORKING REGISTERS.
                                .GLOBL  R1
                                .GLOBL  R2
                                .GLOBL  R3
                                .GLOBL  R4
                                .GLOBL  R5
                                .GLOBL  R6
                                .GLOBL  R7
                                .GLOBL  ERRCD           ; ERROR CODE.
                                .GLOBL  ERDIM           ; DIMENSION ERROR.
                                .GLOBL  ERSUBC          ; SUBSCRIPT ERROR.
                                .GLOBL  ERWSFL          ; NO MORE WORKING ROOM.
                                .GLOBL  DIMCNT          ; DIMENSION/SUBSCRIPT COUNTER.
                                .GLOBL  TPOINT          ; TABLE POINTER.
                                .GLOBL  BYTCNT          ; BYTE ALLOCATION COUNT.
                                .GLOBL  DIMLP           ; DIMENSION LOOP COUNT.
                                .GLOBL  LSP             ; LOW STACK POINTER.
                                .GLOBL  FUDGH           ; STACK FUDGE VALUE (HIGH).
                                .GLOBL  FUDGL           ; STACK FUDGE VALUE (LOW).
                                .GLOBL  DSFLG           ; DIMENSION/SUBSCRIPT FLAG.


                                .GLOBL  VALTG           ; SCALER VALUE TAG.
                                .GLOBL  PAETG           ; POINTER TO ARRAY ELEMENTS.
                                .GLOBL  PNTNTG          ; POINTER TO NAME TABLE-NUMERIC.
                                .GLOBL  PSCTG           ; POINTER TO STRING COUNT.
                                .GLOBL  PNTSTG          ; POINTER TO NAME TABLE-STRING.
                                .GLOBL  RTRNTG          ; RETURN TAG.
                                .GLOBL  LBRKTG          ; LEFT BRACKET TAG.
                                .GLOBL  ITM1TG          ; ITEM 1 TAG.
```

```
                                        ; DIM IS THE DIMENSION TOKEN HANDLER.  IT SETS
                                        ; UP THE SYSTEM TO HANDLE DIMENSIONING BY
                                        ; CLEARING THE BRACKET COUNT AND SETTING THE
                                        ; DIMENSION FLAG IN LCLFLG.


0000  7F  0000G     DIM:  CLR    BRKCNT         ; CLEAR BRACKET COUNT.
0003  86  04              LDA A  DIMFLG,I       ; GET DIMENSION FLAG.
0005  9A  00G             ORA A  LCLFLG,D       ; SET DIM BIT.
0007  97  00G             STA A  LCLFLG,D       ; RESTORE RESULT.
0009  39                  RTS
```

```
                                        ; RPN IS THE RIGHT PARENTHESIS HANDLER.  IT
                                        ; DOES THE COMMON FRONT END FUNCTIONS FOR DIMENSIONING/
                                        ; SUBSCRIPTING AND THEN BRANCHES TO DIMSON IF
                                        ; DIMENSIONING IS REQUIRED OR TO SUBSCR IF
                                        ; SUBSCRIPTING IS REQUIRED.


000A  BD  0000G   RPN:   JSR     PSHRET          ; PUSH RETURN.
000D  7A  0000G          DEC     BRKCNT          ; MATCH LEFT BRACKET.
0010  B6  0000G          LDA A   BRKCNT          ; GET BRACKET COUNT.
0013  27  06             BEQ     RPNCK           ; CHECK FURTHER.
0015  86  FF      SETS:  LDA A   377,I           ; SET UP SUBSCRIPTING.
0017  97  00G            STA A   DSFLG,D
0019  20  09             BRA     RPNFIN          ; FINISH.
0018  96  00G     RPNCK: LDA A   LCLFLG,D        ; GET LOCAL FLAG.
001D  85  04             BIT A   DIMFLG,I        ; EXTRACT DIMENSION FLAG.
001F  27  F4             BEQ     SETS            ; 0=SUBSCRIPTING.
0021  7F  0000G          CLR     DSFLG           ; SET UP DIMENSIONING.
0024  BD  0183'   RPNFIN: JSR     GETVAL          ; GET VALUE/VALUES.
0027  24  03             BCC     GOTV            ; NO ERRORS, CONTINUE.
0029  7E  0000G          JMP     RTRN            ; ERROR RETURN.
002C  30          GOTV:  ISX                     ; SET UP INDEX REGISTER.
002D  A6  00             LDA A   0,X             ; GET TAG.
002F  EE  01             LDX     1,X             ; GET NAME TABLE POINTER.
0031  DF  00G            STX     IPOINT,D        ; SAVE IT.
0033  D6  00G            LDA B   DSFLG,D         ; GET DIM/SUB POINTER.
0035  27  72             BEQ     DIMSON          ; DO DIMENSIONING.
```

```
                                        ; THIS IS THE STARTING POINT FOR SUBSCRIPTING.
                                        ; FOR NUMERICS WITH
                                        ; VALID SUBSCRIPTS, IT RETURNS WITH THE ABSOLUTE
                                        ; POINTER TO THE ARRAY ELEMENT DUELY TAGGED.


0037  A6  04          LDA A  4,X        ; GET ATTRIBUTE.
0039  85  20          BIT A  40,I       ; ARRAY?
003B  26  06          BNE    SUBCAL     ; YES, TRY SUBSCRIPTING
003D  8D  0281'  SUBERR: JSR   ERROR     ; SET SUBSCRIPT ERROR.
0040  7E  0000G  RRET:   JMP   RTRN      ; RETURN.
0043  A6  05     SUBCAL: LDA A  5,X       ; GET ROW VALUE.
0045  E6  06             LDA B  6,X
0047  D0  01G              SUB B  INT1+1,D  ; USERS RANGE OK?
0049  92  00G              SBC A  INT1,D
004B  25  F0              BCS    SUBERR    ; NO, FLAG IT.
004D  DE  00G              LDX    INT1,D    ; BUMP DOWN ROW FOR CALC.
004F  09                   DEX
0050  DF  00G              STX    INT1,D    ; SAVE IT.
0052  DE  00G              LDX    TPOINT,D  ; GET TABLE POINTER.
0054  96  00G              LDA A  DIMCNT,D
0056  26  08              BNE    SUB2      ; CHECK FOR TW SUBSCRIPTS.
0058  EE  07              LDX    7,X       ; GET COLUMN VALUE.
005A  26  F1              BNE    SUBERR    ; NOT 0, HAVE MISMATCH.
005C  08                   INX              ; MAKE = 1
005D  DF  00G              STX    INT2,D    ; SET UP DUMMY COLUMN.
005F  4F                   CLR A            ; NO ACCUMULATE.
0060  5F                   CLR B
0061  20  1F              BRA    SUBFIN    ; FINISH.
0063  EE  07     SUB2:   LDX    7,X       ; GET COLUMN.
0065  27  D6              BEQ    SUBERR    ; NOTHING, MISMATCH.
0067  DE  00G              LDX    TPOINT,D  ; GET TABLE POINTER.
0069  A6  07              LDA A  7,X       ; GET COLUMN VALUE.
006B  E6  08              LDA B  8,X
006D  D0  01G              SUB B  INT2+1,D  ; USERS RANGE OK?
006F  92  00G              SBC A  INT2,D
0071  25  CA              BCS    SUBERR    ; NO, TELL HIM.
0073  DE  00G              LDX    INT2,D    ; BUMP DOWN FOR CALCULATION.
0075  09                   DEX
0076  DF  00G              STX    INT2,D
0078  DE  00G              LDX    TPOINT,D  ; RESTORE TABLE POINTER.
007A  96  00G              LDA A  INT2,D    ; GET COLUMN POINTER.
007C  D6  01G              LDA B  INT2+1,D
007E  EE  07              LDX    7,X       ; SET UP COLUMN DISPLACEMENT.
0080  DF  00G              STX    INT2,D
0082  8D  0279'  SUBFIN: JSR    BYTCAL    ; CALCULATE ADDRESS DISPLACEMENT.
0085  32                   PUL A            ; CLEAN STACK.
0086  32                   PUL A
0087  32                   PUL A
0088  32                   PUL A
0089  32                   PUL A
008A  32                   PUL A
008B  32                   PUL A
008C  96  00G              LDA A  TPOINT,D  ; GET TABLE POINTER.
008E  D6  01G              LDA B  TPOINT+1,D
0090  37                   PSH B            ; STACK TABLE POINTER.
```

```
                0091  36                PSH A
                0092  3E          00G   LDX     TPOINT,D        ; RESET TABLE POINTER.
                0094  A6          08    LDA A   11.,X           ; GET DATA POINTER.
                0096  E6          0C    LDA B   12.,X
                0098  DB          01G   ADD B   R7+1,D          ; CALCULATE ABSOLUTE ELEMENT ADDRESS.
                009A  99          00G   ADC A   R7,D
                009C  37                PSH B                   ; STACK ADDRESS.
                009D  36                PSH A
                009E  A6          08    LDA A   11.,X           ; GET DATA POINTER.
                00A0  E6          0C    LDA B   12.,X
                00A2  37                PSH B                   ; STACK IT.
                00A3  36                PSH A
                00A4  86          00G   LDA A   PAETG,I         ; TAG ARRAY ELEMENT POINTER.
                00A6  36                PSH A
                00A7  20          97    BRA     RRET            ; RETURN.
```

```
                                        ; DIMSON IS THE STARTING POINT FOR
                                        ; DIMENSIONING.  IT BRANCHES TO DIMS FOR
                                        ; STRINGS OR MAINTAINS CONTROL FOR NUMBERS.
                                        ; FOR NUMBERS, IT MUST SELECT THE PATH
                                        ; FOR NEW DIMENSIONING OR RE-DIMENSIONING.


0089  81  006   DIMSON: CMP A  PNTSTG,I  ; IS IT STRING?
008B  26  03            BNE    B6       ; NO, DO NUMBER.
008D  7E  013F'         JMP    DIMS     ; YES, DO STRING.
0090  86  09    B6:     LDA A  9,I      ; CLEAN STACK.
0092  BD  0297'         JSR    CLEANS
0095  DE  006           LDX    TPOINT,D ; GET TABLE POINTER.
0097  A6  04            LDA A  4,X      ; GET ATTRIBUTE.
0099  81  40            CMP A  100,I     ; SIMPLE DEFINED?
009B  26  05            BNE    DIMCAL   ; NO, GO DIMENSION.
009D  BD  0281' DIMERR: JSR    ERROR    ; SET DIMENSION ERROR.
00A0  20  30            BRA    DR       ; RETURN.
00A2  96  006   DIMCAL: LDA A  DIMCNT,D  ; GET DIMENSION COUNT.
00A4  26  05            BNE    CA       ; NOT 0, MUST ALREADY HAVE 2
00A6  CE  0001          LDX    1,I      ; FORCE SECOND = 1 FOR MULT.
00A9  DF  006           STX    INT2,D
00AB  4F        CA:     CLR A           ; CLEAR ACCUMULATORS.
00AC  5F                CLR B
00AD  BD  027A'         JSR    BYTCAL   ; CALCULATE BYTE COUNT.
00B0  25  20            BCS    DR       ; OVER 16 BITS WORTH
00B2  DE  006           LDX    R7,D     ; SAVE RESULT.
00B4  DF  006           STX    BYTCNT,D
00B6  DE  006           LDX    TPOINT,D ; GET TABLE POINTER.
00B8  A6  04            LDA A  4,X      ; GET ATTRIBUTE.
00BA  85  20            BIT A  40,I     ; IS IT ARRAY.
00BC  27  17            BEQ    B7       ; NO, DO DIMENSION
                                        ; START REDIMENSION
                                        ; CHECK HERE.
00BE  A6  09            LDA A  9,X      ; GET ORIGINAL BYTE COUNT.
00C0  E6  0A            LDA B  10.,X
00C2  D0  01G           SUB B  BYTCNT+1,D ; COMPARE TO NEW BYTE COUNT.
00C4  92  006           SBC A  BYTCNT,D
00C6  25  D5            BCS    DIMERR   ; NEW BYTE COUNT TOO BIG.
00C8                    SEI              ; PROTECT SYSTEM.
00C8  01  0E    BYTE    01,17
00CA  BD  022F'         JSR    SETDIM   ; SET UP NEW SUBSCRIPT LIMITS.
00CD  86  20            LDA A  40,I     ; GET ATTRIBUTE.
00CF  A7  04            STA A  4,X      ; CLEAR ALLOK BIT.
00D1  0E                CLI              ; PROTECTION NO LONGER NEEDED.
00D2  7E  0000G DR:     JMP    RTRN     ; RETURN.
                                        ; START INITIAL (NEW)
                                        ; DIMENSION HERE.
00D5  BD  01FA' B7:     JSR    ALOCAT   ; CHECK SPACE
00D8  25  E8            BCS    DR       ; 1 = ALLOCATE CK FAILED
00DA  4F        DOALC:  CLR A           ; CALCULATE STARTING DATA ADDRESS.
00DB  C6  05            LDA B  5,I
00DD  D8  01G           ADD B  LSP+1,D
00DF  99  006           ADC A  LSP,D
0101  97  006           STA A  R7,D     ; SAVE IT.
0103  D7  01G           STA B  R7+1,D
0105                    SEI              ; PROTECT SYSTEM.
```

```
                0105    01      0F                      .BYTE   01,17
    57          0107    BD      0000G                   JSR     DISBLE          ; DISABLE BREAKS.
    58          010A    DE      00G                     LDX     TPOINT,D        ; GET TABLE POINTER.
    59          010C    86      20                      LDA A   40,I            ; GET ARRAY ATTRIBUTE.
    60          010E    A7      04                      STA A   4,X             ; SET ARRAY ATTRIBUTE.
    61          0110    BD      022F'                   JSR     SETDIM          ; SET UP SUBSCRIPT LIMITS.
    62          0113    96      00G                     LDA A   BYTCNT,D        ; SET UP BYTE COUNT IN TABLE.
    63          0115    A7      09                      STA A   9.,X
    64          0117    96      01G                     LDA A   BYTCNT+1,D
    65          0119    A7      0A                      STA A   10.,X
    66          011B    BD      025F'                   JSR     SETLSP          ; SET UP LSP & HEADER.
    67          011E    86      10                      LDA A   20,I            ; GET ARRAY ATTRIBUTE FOR HEADER.
    68          0120    A7      02                      STA A   2,X             ; SET HEADER ATTRIBUTE.
    69          0122    0E                              CLI                     ; ENABLE INTERRUPTS.
    70          0123    FE      0000G                   LDX     R7              ; GET START ADDRESS.
    71          0126    86      40                      LDA A   100,I           ; GET UNDEFINED BYTE.
    72          0128    A7      00              LP:     STA A   0,X             ; STORE IT IN DATA FIELD.
    73          012A    DE      00G                     LDX     DIMLP,D         ; GET LOOP COUNT.
    74          012C    09                              DEX                     ; BUMP DOWN.
    75          012D    27      0B                      BEQ     DRFIN           ; 0 = FINISHED.
    76          012F    DF      00G                     STX     DIMLP,D         ; SAVE IT.
    77          0131    DE      00G                     LDX     R7,D            ; ADD 8 TO DATA ADDRESS.
    78          0133    BD      0000G                   JSR     A8X
    79          0136    DF      00G                     STX     R7,D            ; SAVE IT.
    80          0138    20      EE                      BRA     LP              ; CONTINUE LOOP.
    81          013A    BD      0000G           DRFIN:  JSR     ENABLE          ; ENABLE BREAKS.
    82          013D    20      B3                      BRA     DR              ; FINISHED, RETURN.
```

```
                                        ; DIMS DIMENSIONS NEW STRING OR
                                        ; RE-DIMENSIONS OLD STRINGS. FOR NEW
                                        ; DIMENSIONING, IT CALLS DIMSTR. FOR
                                        ; RE-DIMENSIONING, IT CHECKS TO MAKE
                                        ; SURE A SPACE OVERFLOW CONFLICT HAS
                                        ; NOT OCCURED AND CHECK TO IF CURRENT
                                        ; LENGTH NEEDS TO BE PULLED BACK.

013F  86  05      DIMS:  LDA A  5,I     ; CLEAR TABLE ADDRESS FROM STACK.
0141  BD  0297'          JSR   CLEANS
0144  DE  00G            LDX   TPOINT,D  ; GET TABLE POINTER.
0146  A6  04            LDA A  4,X      ; GET ATTRIBUTE.
0148  85  80            BIT A  200.I    ; DIMENSIONED?
014A  27  04            BEQ    B12      ; YES, TRY RE-DIMENSION.
014C  8D  3C            BSR    DIMSTR   ; NO, DO FIRST DIMENSION.
014E  20  28            BRA    BkET     ; RETURN
0150  96  00G    B12:   LDA A  INT1,D    ; GET NEW DIMENSION.
0152  D6  01G           LDA B  INT1+1,D
0154  CB  04            ADD B  4.I      ; ADD IN HEADER CONSTANT-1
0156  89  00            ADC A  0,I
0158  25  28            BCS    DMER     ; OVER 16 BITS, ERROR EXIT.
015A  EE  08            LDX    11.,X    ; GET OLD BYTE COUNT POINTER
015C  E0  01            SUB B  1,X      ; COMPARE (NEW-OLD).
015E  A2  00            SBC A  0,X
0160  24  23            BCC    DMER     ; NEW IS LARGER THAN OLD, ERROR
0162  BD  0000G          JSR   DISBLE    ; DISABLE BREAKS.
0165  96  00G            LDA A  INT1,D   ; GET NEW DIMENSION.
0167  D6  01G           LDA B  INT1+1,D
0169  DE  00G            LDX   TPOINT,D  ; GET TABLE POINTER.
016B  A7  05            STA A  5,X      ; SET NEW DIMENSION IN TABLE.
016D  E7  06            STA B  6,X
016F  E0  08            SUB B  8.,X     ; SUBTRACT FROM CURRENT COUNT.
0171  A2  07            SBC A  7,X
0173  25  06            BCS    B11      ; CURRENT LARGER?
0175  BD  0000G  CBRET: JSR    ENABLE   ; ENABLE BREAKS.
0178  7E  0000G  BRET:  JMP    RTRN     ; NO, RETURN.
017B  A6  05     B11:   LDA A  5,X      ; YES, MAKE CURRENT = NEW.
017D  E6  06            LDA B  6,X
017F  A7  07            STA A  7,X
0181  E7  08            STA B  8.,X
0183  20  F0            BRA    CBRET    ; RETURN.
0185  BD  02A1'  DMER:  JSR    ERROR    ; SET ERROR FLAG.
0188  20  EE            BRA    BRET     ; RETURN.
```