```
                                ;                               1 BYTE PRIMERY ADDRESS FOR ROM PRINT
                                ;
                                ;       CRT
                                ;
058A                            Z       TMP1,1          ;TMP1,2,3 MUST BE IN ORDER
00A5                            Z       TMP2,1
00A6                            Z       TMP3,1
00A7                            Z       TMPHY,1
00A8                            Z       TMPLY,1
00A9                            Z       TMPHX,1
00AA                            Z       TMPLX,1
00AB                            Z       COLCT,1
00AC                            Z       ROWCT,1
00AD                            Z       BLINK,1
00AE                            Z       DOTON,1
00AF                            Z       FULL,1
00B0                            Z       DISCNT,1
00B1                            Z       CURSOR,1
00B2                            Z       FONT,1          ; DISPLAY FONT TYPE
00B3                            N       PGCTR,2         ; FOR SCREEN CLEAN UP
058C                            N       FULSTT,1        ;PAGE FULL STATUS
058E                            N       LINELN,1        ;LINE SIZE FOR DVST
                                ;
                                ;       MAG TAPE & IEC
                                ;
058F                            N       MTERR,1         ; MAG TAPE ERROR COUNT
0590                            Z       MTX8YT,1        ; MAG TAPE EXTRA BYTES/REC  COUNT
00B4                            Z       RECCNT,2        ; RECORDS AVAILABLE ON  PRESENT FILE
00B5                            Z       EARLYW,1
00B7                            Z       CCODES,1
00B8                            Z       MTFLGS,1        ;FLAGS
00B9                            Z       BPIAMTB,1       ;BUFFER FOR PERIPHERAL INTERFACE REGISTER
00BA                            Z       FILLOC,2        ;CURRENT FILE LOCATION (BLOCK NUMBER)
00BB                            Z       FILFND,2        ;FILE (BLOCK) NUMBER TO BE FOUND
                                ;
                                ;       KEY BOARD
                                ;
00BD                            N       CNTL,1          ;CONTROL KEY FLAG
0591                            N       TTY2,1          ;TTY LOCK CONSTANT
0592                            N       SHIFT1,1        ;SHIFT KEY CONSTANT 1
0593                            N       SHIFT2,1        ;SHIFT KEY CONSTANT 2
0594                            N       ROW,1           ;ROW OF MATRIX
0595                            N       COL,1           ;COLUMN OF SWITCH MATRIX
0596                            N       STKUSE,1        ;STACK USAGE COUNTER
0597                            N       KEYTIM,1        ;KEY DELAY TIME REGISTER
0598                            N       KEYCNT,1        ;ACTUAL DELAY PASSED FOR KEY0
0599                            N       KBMRK,2         ;TEMP  STACK POINTER
059A                            N       KBSTK,8         ;KEY STACK ---KEY0 IS FIRST ENTRY
059C                            N       ACCEL,1         ;REPEAT SPEED CONTROL
05A4                            N       KYRATE,1        ;RATE CONTROLS
05A5                            N       VLDKEY,1        ;VALID KEY
                                ;
                                ;       F. P. PACK
                                ;
05A6                            Z       T1,1            ;COUNTERS
00BF                            Z       T2,1
00C0                            Z       T3,1
```

```
00C1            Z       T4,1
00C2            Z       SIGNS,1
00C3            Z       TABPNT,2        ;POINTERS INTO TABLES
00C4            Z       MULPNT,2
00C6            Z       Y5,1            ;48 BIT MANTISSA
00C8            Z       Y4,1
00C9            Z       Y3,1
00CA            Z       Y2,1
00CB            Z       Y1,1
00CC            Z       Y0,1
00CD            Z       X5,1            ;48 BIT MANTISSA
00CE            Z       X4,1
00CF            Z       X3,1
00D0            Z       X2,1
00D1            Z       X1,1
00D2            Z       X0,1
00D3            Z       DEXP,2
00D4            ASGN    EXH,DEXP        ;FOR GOOD OLD JOHN
00D4            ASGN    EXL,DEXP+1
00D5            N       DIGFLG,1        ;ZERO IF VALID DIGITS IN ASCII TO FPN
05A7            N       ITSINT,1        ;0=IT IS NOT AN INTEGER (ASCII TO FPN)
05A8            N       EFLAG,1         ;SO LITERAL 'E' SEEN
05A9            N       FPA,8           ;TEMP FPN A
05AA            N       FPB,8           ;TEMP FPN B
05B2            N       FPC,8           ;TEMP FPN C
05BA            N       FUZA,1          ;BITS THAT MUST BE EQUAL
05C2            N       FUZB,8          ;WIDTH OF ZERO
05C3            N       FUZR,1          ;CURRENT FUZZIENESS
05CB            N       CONV1,2         ;PTR TO ANGLE CONVERSION CONSTANT
05CC            N       DETADD,8        ;DETERMINATE
05CE            N       KERNEL,8        ;RANDOM NUMBER SEED
              ;
              ;         FORMAT CONTROL VARIABLES
              ;
05D6            Z       ISP,2           ; IMAGE STRING POINTER
00D6            N       ISB,2           ; BASE FOR IMAGE STRING
05DE            Z       DP,2            ; DATA POINTER
00D8            N       DB,2            ;DATA BASE POINTER
05E0            Z       L1,1            ; REPEAT LOOP 1
00DA            Z       SP1,2           ; RELATIVE STRING POINTER 1
00DB            Z       L2,1            ; REPEAT LOOP 2
00DD            Z       SP2,2           ; RELATIVE STRING POINTER 2
00DE            Z       L3,1            ; REPEAT LOOP 3
00E0            Z       SP3,2           ; RELATIVE STRING POINTER 3
00E1            Z       L4,1            ; REPEAT LOOP 4
00E3            Z       SP4,2           ; RELATIVE STRING POINTER 4
00E4            Z       CURSER,2        ; RELATIVE CUSER POINTER
00E6            Z       PARCNT,1        ; PARENTHESIS COUNT
00E8            Z       ISL,2           ; IMAGE STRING LENGTH
00E9            Z       DL,2            ; DATA LENGTH
00EB            Z       DT,1            ; DATA TYPE
00ED            Z       LITFLG,1        ; LITERAL FLAG
00EE            Z       LRSFLG,1        ; LRS SUPPRESS FLAG
00EF            Z       NUMFLG,1        ; NUMBER FLAG
00F0            Z       FFLG,1          ; 9S REQUIRED FLAG
00F1            Z       PFLG,1          ; PLUS FLAG
00F2            Z       MFLG,1          ; MINUS FLAG
```

```
00F3        Z     DOLFLG,1      ; DOLLAR SIGN FLAG
00F4        Z     COMFLG,1      ; COMMA FORMAT FLAG
00F5        Z     FMTFLG,1      ; FORMAT TYPE IN PROCESS
00F6        Z     FLAGS,1       ; CONDENSED FLAGS
00F7        ASGN  K0,R0         ; K0-K6 ARE TEMPORARY WORKING LOCATIONS
0000        ASGN  K1,R0+1
0001        ASGN  K2,R1
0002        ASGN  K3,R1+1
0003        ASGN  K4,R2
0004        ASGN  K5,R2+1
0005        ASGN  K6,R3
0006        N     ES,1          ; EXPONENT SIGN
05E2        N     E1,1          ; EXPONENT LSD
05E3        N     E2,1
05E4        N     E3,1          ; EXPONENT MSD
05E5        N     N1,10         ; NUMBER LSD'S
05E6        N     N11,1
05F0        N     N12,1         ; NUMBER MSD
05F1        N     NS,1          ; NUMBER SIGN
05F2        N     DDP,1         ; DIGITS DECIMAL POINT
05F3        N     DN1,1         ; NUMBER OF LEFT DIGIT
05F4        N     DN2,1         ; NUMBER OF RIGHT DIGITS
05F5        N     DIGCNT,1      ; NUMBER OF OUTPUT DIGITS
05F6        N     F1,1          ; AS REQUIRED LEFT
05F7        N     F2,1          ; AS REQUIRED RIGHT
05F8        N     EXP,1         ; CONDENSED EXPONENT
05F9        N     EXS,1         ; EXPONENT BINARY SIGN
05FA        N     LSPS,1        ; LEADING SPACES
05FB        N     LDIG,1        ; LEFT DIGIT COUNT
05FC        N     LDIGA,2       ; LEFT DIGIT BEGINNING ADDRESS
05FD        N     LCOM,1        ; LEADING DIGIT COUNT (REMAINDER)
05FF        N     ZERO1,1       ; LEFT TRAILING ZEROS
0600        N     ZERO2,1       ; RIGHT LEADING ZERO COUNT
0601        N     RDIG,1        ; RIGHT PRINTING DIGITS COUNT
0602        N     RDIGA,2       ; RIGHT DIGIT BEGINNING ADDRESS
0603        N     ZERO3,1       ; RIGHT TRAILING ZERO COUNT
0605        N     ZERCNT,1      ; COMMON ZERO COUNT
0606        N     ZERSVH,1      ; ZERO SAVE ADDRESS HIGH
0607        N     ZERSVL,1      ; ZERO SAVE ADDRESS LOW
0608        N     COMCNT,1      ; COMMA COUNT
0609        N     DPCONT,1      ; DEFAULT CONTROL
060A        N     TABCNT,1      ; TAB COUNT - CONTROL
            ;
            ;     DIMENSION AND SUBSCRIPT
            ;
0603        ASGN  INT1,R11      ;ROW SIZE
0016        ASGN  INT2,R12      ;COLUMN SIZE
0018        ASGN  DIMCNT,R13    ;COUNTER
001A        ASGN  TPOINT,R14    ;TABLE ENTRY POINTER
001C        ASGN  BYTCNT,R15    ;ALLOCATION BYTE COUNT
001E        ASGN  DIMLP,R16     ;LOOP COUNTER
0020        ASGN  DSFLG,R17     ;DIMENSION/SUBSCRIPT FLAG
0022        ASGN  RETH,R18      ;TEMP RETURN POINT
0024        ASGN  RETL,R18+1
            ;
            ;
            ;******************************************************************
```

```
472  0025                        Z       IOSNTP,2        ;I/O SYSTEM NAME TABLE POINTER
473                              ;*******************************************************
474                              ;
475                              ;
476  00E6                        ASGN    PGZEND,PGZ
477  00FA                        ASGN    PGNEND,PGN
478                              ;
479          0001'               .END
```

SYMBOL TABLE

```
ACCEL =  05A4 G   ADBANK= XXXXXX G   ADMAIN= XXXXXX G   ALLTG =  0014 G   ANCRNT=  0066 G
ANHOLD=  0462 G   ANLINCR=  0068 G   ANYINT=  0508 G   APPEND= XXXXXX G   ATSNTG=  0012 G
AXIS  = XXXXXX G   A.END =  0095 G   A.MAX =  0093 G   A.PRIM=  009G G   A.PTR =  0098 G
A.SEC =  0097 G   A.STAT=  008F G   A.STRT=  0091 G   BAKSTG=  0013 G   BANK  =  007C G
BINCTL= XXXXXX G   BLINK =  008E G   BNKMAP= XXXXXX G   BPIANT=  008A G   BRKCNT=  043D G
BSTMT = XXXXXX G   BYTCNT=  001E G   CALL  = XXXXXX G   CALLTG=  0017 G   CCODES=  0088 G
CDOPTR=  043B G   CDSPTR=  0439 G   CHAR  =  008D G   CHRCNT=  0088 G   CHRVCT=  0473 G
CLOSE = XXXXXX G   CLPTR =  0040 G   CNT   =  0031 G   CNTL  =  0591 G   COL   =  0596 G
COLCNT=  009E G   COLCT =  00AC G   COMCNT=  0609 G   COMFLG=  00F5 G   COMP  =  003A G
COMPR = XXXXXX G   CONV1 =  05CC G   COSTHA=  04E2 G   CRASH =  041A G   CROSS = XXXXXX G
CRSTAT=  006A G   CRTDRV= XXXXXX G   CTKN  =  0053 G   CURSER=  00E6 G   CURSOR=  0082 G
DB    =  05E0 G   DDP   =  05F3 G   DELETE= XXXXXX G   DETADD=  05CE G   DEXP  =  0004 G
DIGCNT=  05F6 G   DIGFLG=  05A7 G   DIMCNT=  001A G   DIMLP =  0020 G   DIMSUB= XXXXXX G
DISCNT=  0081 G   DL    =  0058 G   DLTXS =  0490 G   DLTYS =  0420 G   DN1   =  05E4 G
DN2   =  05F5 G   DOLFLG=  00F4 G   DOTON =  00AF G   DP    =  0008 G   DPCONT=  060A G
DREXTA=  0058 G   DREXTB=  0058 G   DSFLG =  0022 G   DSPOUT= XXXXXX G   DSPSTT=  0071 G
DSPVCT=  0460 G   DT    =  00ED G   EARLYW=  0087 G   EDTBFR=  0221 G   EDTEND=  0060 G
EDTFLG=  0478 G   EDTMAX=  0064 G   EDTPTR=  0062 G   EFLAG =  05A9 G   ENDTBL=  041A G
EOFTBL=  03C9 G   EOLCHK=  0078 G   EOLTG =  0018 G   EOSTG =  0019 G   EQU   =  0030 G
ERCTRA=  0435 G   ERCTRB=  0436 G   ERRCD =  0048 G   ERRCOB=  004C G   ES    =  05E2 G
ESTG  =  0002 G   ETXCHR=  0079 G   EVLEN = XXXXXX G   EVLSUB= XXXXXX G   EVLTBL= XXXXXX G
EXH   =  0004 G   EXL   =  0005 G   EXP   =  05F9 G   EXS   =  05FA G   E1    =  05E3 G
E2    =  05E4 G   E3    =  05E5 G   FFLG  =  00F1 G   FILES = XXXXXX G   FILEND=  008D G
FILLOC=  0088 G   FLAG  =  0032 G   FLAGS =  00F7 G   FLCTRL= XXXXXX G   FLDATA= XXXXXX G
FMTFLG=  00F6 G   FNTBL =  03DD G   FONT  =  0083 G   FORTG =  0004 G   FPA   =  05AA G
FPB   =  0582 G   FPC   =  0588 G   FPMATH= XXXXXX G   FUDGH =  0001 G   FUDGL =  00E4 G
FULL  =  0080 G   FULSTT=  058E G   FUZA  =  05C2 G   FUZB  =  05C3 G   FUZR  =  05CB G
F1    =  05F7 G   F2    =  05F8 G   GETKEY= XXXXXX G   GETLIN= XXXXXX G   GLBFLG=  0055 G
GNSTG =  0003 G   GRAF  = XXXXXX G   GRAPH = XXXXXX G   HCTRCS=  0388 G   HEP   =  003E G
HLDEOF=  0434 G   IDLVCT=  0470 G   IECDRV= XXXXXX G   IECRLF=  0463 G   IMXTG =  0007 G
INFO  = XXXXXX G   INPCTL= XXXXXX G   INPNT =  0098 G   INTQQQ=  005E G   INTSRV= XXXXXX G
INT1  =  0016 G   INT2  =  0018 G   IOBER1=  0268 G   IOCTL = XXXXXX G   IOFLGS=  008A G
]]]]]
IOFUNC=  008E G   IOKONS= XXXXXX G   IOPROC= XXXXXX G   IOSNTP=  00F8 G   IOSYSF=  0467 G
ISR   =  050E G   ISL   =  00E9 G   ISP   =  00D6 G   ITM1TG=  00DE G   ITM2TG=  00DF G
ITSINT=  05A8 G   JMPAX =  0450 G   JMPX  =  043E G   KBFLAG=  0068 G   KBIN  =  0076 G
KBMRK =  059A G   KBPROC= XXXXXX G   KBSTK =  059C G   KERNEL=  0506 G   KEYCNT=  0599 G
KEYDRV= XXXXXX G   KEYQUE=  0100 G   KEYSTK=  0411 G   KEYTIM=  0598 G   KYRATE=  0585 G
K0    =  0000 G   K1    =  0001 G   K2    =  0002 G   K3    =  0003 G   K4    =  0004 G
K5    =  0005 G   K6    =  0006 G   LBRKTG=  0010 G   LCLFLG=  0054 G   LCOM  =  05FF G
LDAX  =  0448 G   LDBX  =  0451 G   LDIG  =  05EC G   LDIGA =  05ED G   LDXX  =  0445 G
LENGTH=  009A G   LEX   = XXXXXX G   LEXCNT=  0036 G   LINELN=  058F G   LISTTG=  0005 G
LITFLG=  00EE G   LNNOTG=  000D G   LRSFLG=  00EF G   LSP   =  0045 G   LSPS  =  05FB G
LSTKEY=  006C G   LUNNO =  007D G   L1    =  000A G   L2    =  000D G   L3    =  00E0 G
L4    =  00E3 G   MAGEND=  006F G   MASK  =  0039 G   MATOPR= XXXXXX G   MFLG  =  00F3 G
MSKCTR=  050A G   MTBFR =  011F G   MTDRV = XXXXXX G   MTERR =  0590 G   MTFLGS=  0089 G
MTMASK=  0468 G   MTMAX =  0083 G   MTPTR =  0085 G   MTSREG=  0081 G   MTSTAT=  0082 G
MTSTT2=  0070 G   MTXBYT=  0084 G   MULPNT=  00C6 G   NLPTR =  004F G   NOOUT =  0087 G
NS    =  05F2 G   NTPTR =  0051 G   NULCHR=  007A G   NULLTG=  0000 G   NUMFLG=  00F0 G
N1    =  05E6 G   N11   =  05F0 G   N12   =  05F1 G   ONT9L =  0389 G   OPRADR=  0056 G
OPRRTN=  0437 G   OUTCTL= XXXXXX G   PAETG =  0008 G   PARCNT=  00E8 G   PARSE = XXXXXX G
PCHAR = XXXXXX G   PFLG  =  00F2 G   PGCTR =  058C G   PGMEVL= XXXXXX G   PGMLIS= XXXXXX G
PGMPTR=  003D G   PGMTG =  0006 G   PGN   =  060C     PGNEND=  060C G   PGZ   =  00FA
PGZEND=  00FA G   PIAEND=  058A G   PIATBL=  050C G   PLOSTG=  0001 G   PNDEOF=  006E G
PNDFLG=  006D G   PNTNTG=  000A G   PNTSTG=  0008 G   POINT =  009C G   POINTB=  0476 G
PPEOF =  0465 G   PPEOR =  0464 G   PPMODE=  0077 G   PPNUL =  0466 G   PRINTF= XXXXXX G
PRTERR= XXXXXX G   PRTTG =  0016 G   PSCTG =  0009 G   PWRUP = XXXXXX G   QEND  =  011E G
QIN   =  0072 G   QOUT  =  0074 G   RDIG  =  0602 G   RDIGA =  0603 G   RECCNT=  0085 G
```

```
RAMMAP PAGE 0 AND ABOVE SPACE  RT-11 MMAC  VM02-10   14-OCT-76 01:40:43 PAGE 1+
.SYMBOL TABLE

RECNO = 007E G      RENUMB= xxxxxx G      RETH  = 0024 G      RETL  = 0025 G      ROW   = 0595 G
ROWCT = 008D G      RTRNTG= 0016 G      R0    = 0000 G      R1    = 0002 G      R10   = 0014 G
R11   = 0016 G      R12   = 0018 G      R13   = 001A G      R14   = 001C G      R15   = 001E G
R16   = 0020 G      R17   = 0022 G      R18   = 0024 G      R19   = 0026 G      R2    = 0004 G
R20   = 0028 G      R21   = 002A G      R22   = 002C G      R23   = 002E G      R3    = 0006 G
R4    = 0008 G      R5    = 000A G      R6    = 000C G      R7    = 000E G      R8    = 0010 G
R9    = 0012 G      SBP   = 0043 G      SCMSED= 0479 G      SCORE = 0038 G      SCRTCH= 0285 G
SEMITG= 0011 G      SHIFT1= 0593 G      SHIFT2= 0594 G      SHNTCT= 0033 G      SHUNT = xxxxxx G
SIGNS = 00C3 G      SINTHA= 040A G      SLOPH = 0000 G      SLOPL = 00E4 G      SP1   = 0008 G
SP2   = 00DE G      SP3   = 00E1 G      SP4   = 00E4 G      STAT3?= 0078 G      STAX  = 0457 G
STKUSE= 0597 G      STPTR = 0038 G      STRING= xxxxxx G      SYMTAB= xxxxxx G      TABCNT= 060B G
TABPNT= 00C4 G      TABPTR= 00A0 G      TAGPTR= 0088 G      TAPE  = xxxxxx G      TCOL  = 00A2 G
TEMPX = 04F2 G      TEMPY = 0502 G      TFLGS1= 0035 G      TMPHX = 00A? G      TMPHY = 00A8 G
TMPLX = 00A6 G      TMPLY = 00A9 G      TMP1  = C?05 G      TMP2  = 00A6 G      TMP3  = 00A7 G
TPOINT= 001C G      TRANSL= xxxxxx G      TRASH = 011E G      TTY2  = 0592 G      TYPIN = xxxxxx G
T1    = 00BF G      T2    = 00C0 G      T3    = 00C1 G      T4    = 00C2 G      UNCOMP= xxxxxx G
UNLEX = xxxxxx G      USING = xxxxxx G      USRORG= 0047 G      UTILS = xxxxxx G      VOLTG = 000C G
VLOKEY= 05A6 G      XAXIS = 0080 G      XEQSP = 0049 G      XEQ   = 0424 G      XFNBNK= 0469 G
XFNMAP= xxxxxx G      XLAST = 04C2 G      XMAXW = 0482 G      XMINS = 04A2 G      XMINW = 047A G
XNF4  = 04EA G      XSF   = 04A8 G      X0    = 0003 G      X1    = 0002 G      X2    = 0001 G
X3    = 00D0 G      X4    = 00CF G      X5    = 00CE G      YAXIS = 00A4 G      YLAST = 04D2 G
YMAXW = 0492 G      YMINS = 0482 G      YMINW = 048A G      YNEW  = 04FA G      YSF   = 04CA G
Y0    = 00CD G      Y1    = 00CC G      Y2    = 00CB G      Y3    = 00CA G      Y4    = 00C9 G
Y5    = 00C8 G      ZERCNT= 0606 G      ZERO1 = 0600 G      ZERO2 = 0601 G      ZERO3 = 0605 G
ZERSVH= 0607 G      ZERSVL= 0608 G      ZX    = 0041 G
  ABS.   060C    00
         0000    01
ERRORS DETECTED:  0      WARNINGS POSTED:  0      FREE CORE:  1729.  WORDS
.SY:RAMMAP/C<DK1:SEICLI,RAMMAP
```

CROSS REFERENCE TABLE (CREF V01-03)

	1-65#	1-66#	1-67#	1-68#	1-69#	1-70#	1-71#	1-72#	1-73#	1-74#	1-75#	1-76#	1-77#	1-78#
	1-79#	1-80#	1-81#	1-82#	1-83#	1-84#	1-85#	1-86#	1-87#	1-88#	1-89#	1-90#	1-94#	1-95#
	1-97#	1-98#	1-102#	1-103#	1-104#	1-105#	1-106#	1-107#	1-108#	1-109#	1-110#	1-111#	1-112#	1-113#
	1-114#	1-115#	1-116#	1-117#	1-118#	1-119#	1-120#	1-121#	1-122#	1-123#	1-124#	1-125#	1-129#	1-130#
	1-131#	1-132#	1-133#	1-134#	1-135#	1-136#	1-140#	1-141#	1-142#	1-143#	1-144#	1-148#	1-149#	1-150#
	1-151#	1-152#	1-153#	1-154#	1-155#	1-156#	1-157#	1-158#	1-159#	1-160#	1-161#	1-163#	1-165#	1-166#
	1-167#	1-168#	1-169#	1-170#	1-171#	1-172#	1-173#	1-177#	1-178#	1-179#	1-180#	1-181#	1-182#	1-183#
	1-184#	1-185#	1-186#	1-188#	1-189#	1-190#	1-194#	1-195#	1-196#	1-197#	1-198#	1-199#	1-203#	1-204#
	1-205#	1-206#	1-207#	1-208#	1-209#	1-210#	1-211#	1-212#	1-213#	1-214#	1-215#	1-216#	1-217#	1-218#
	1-222#	1-223#	1-224#	1-225#	1-226#	1-227#	1-228#	1-229#	1-230#	1-231#	1-232#	1-233#	1-234#	1-235#
	1-236#	1-237#	1-238#	1-239#	1-240#	1-242#	1-245#	1-246#	1-247#	1-248#	1-249#	1-250#	1-251#	1-252#
	1-253#	1-254#	1-255#	1-256#	1-257#	1-258#	1-259#	1-260#	1-261#	1-262#	1-263#	1-264#	1-265#	1-266#
	1-267#	1-268#	1-269#	1-270#	1-271#	1-272#	1-276#	1-277#	1-278#	1-279#	1-280#	1-281#	1-282#	1-283#
	1-284#	1-285#	1-286#	1-287#	1-288#	1-289#	1-290#	1-291#	1-292#	1-293#	1-294#	1-295#	1-296#	1-297#
	1-305#	1-306#	1-307#	1-308#	1-309#	1-310#	1-311#	1-312#	1-313#	1-314#	1-315#	1-316#	1-317#	1-318#
	1-319#	1-320#	1-321#	1-322#	1-326#	1-327#	1-328#	1-329#	1-330#	1-331#	1-332#	1-333#	1-334#	1-338#
	1-339#	1-340#	1-341#	1-342#	1-343#	1-344#	1-345#	1-346#	1-347#	1-348#	1-349#	1-350#	1-351#	1-355#
	1-356#	1-357#	1-358#	1-359#	1-360#	1-361#	1-362#	1-363#	1-364#	1-365#	1-366#	1-367#	1-368#	1-369#
	1-370#	1-371#	1-372#	1-373#	1-374#	1-375#	1-376#	1-377#	1-378#	1-379#	1-380#	1-381#	1-382#	1-383#
	1-384#	1-385#	1-386#	1-387#	1-388#	1-392#	1-393#	1-394#	1-395#	1-396#	1-397#	1-398#	1-399#	1-400#
	1-401#	1-402#	1-403#	1-404#	1-405#	1-406#	1-407#	1-408#	1-409#	1-410#	1-411#	1-412#	1-413#	1-414#
	1-415#	1-416#	1-417#	1-418#	1-419#	1-420#	1-421#	1-422#	1-423#	1-424#	1-425#	1-426#	1-427#	1-428#
	1-429#	1-430#	1-431#	1-432#	1-433#	1-434#	1-435#	1-436#	1-437#	1-438#	1-439#	1-440#	1-441#	1-442#
	1-443#	1-444#	1-445#	1-446#	1-447#	1-448#	1-449#	1-450#	1-451#	1-452#	1-453#	1-454#	1-455#	1-456#
	1-460#	1-461#	1-462#	1-463#	1-464#	1-465#	1-466#	1-467#	1-468#	1-472#	1-476#	1-477#		
A.END	1-261#	1-261#												
A.MAX	1-260#	1-260#												
A.PRIM	1-258#	1-258#												
A.PTR	1-263#	1-263#	1-264	1-264										
A.SEC	1-262#	1-262#												
A.STAT	1-257#	1-257#												
A.STRT	1-259#	1-259#												
ACCEL	1-349#	1-349#												
ADBANK	1-19#													
]]]]]														
ADMAIN	1-19#													
ALLTG	1-85#	1-85#												
ANCRNT	1-207#	1-207#												
ANHOLD	1-209#	1-209#												
ANINCR	1-208#	1-208#												
ANYINT	1-295#	1-295#												
APPEND	1-21#													
ATSNTG	1-83#	1-83#												
AXIS	1-29#													
BAKSTG	1-84#	1-84#												
BANK	1-235#	1-235#												
BINCTL	1-26#													
BLINK	1-314#	1-314#												
BNKMAP	1-23#													
BPTAMT	1-332#	1-332#												
BRKCNT	1-190#	1-190#												
BSTMT	1-22#													
BYTCNT	1-464#	1-464#												
CALL	1-25#													
CALLTG	1-88#	1-88#												
CCODES	1-330#	1-330#												
CDOPTR	1-189#	1-189#												

CROSS REFERENCE TABLE (CREF V01-03)

CDSPTR	1-188#	1-188#					
CHAR	1-255#	1-255#					
CHRCNT	1-253#	1-253#					
CHRVCT	1-245#	1-245#					
CLOSE	1-28#						
CLPTR	1-177#	1-177#					
CNT	1-149#	1-149#					
CNTL	1-338#	1-338#					
COL	1-343#	1-343#					
COLCNT	1-267#	1-267#					
COLCT	1-312#	1-312#					
COMCNT	1-454#	1-454#					
COMFLG	1-416#	1-416#					
COMP	1-156#	1-156#					
COMPR	1-21#						
CONV1	1-386#	1-386#					
COSTHA	1-289#	1-289#					
CRASH	1-157#	1-157#					
CROSS	1-29#						
CRSTAT	1-210#	1-210#					
CRTDRV	1-30#						
CTKN	1-180#	1-180#					
CURSER	1-404#	1-404#					
CURSOR	1-318#	1-318#					
DB	1-395#	1-395#					
DDP	1-434#	1-434#					
DELETE	1-21#						
DETADD	1-387#	1-387#					
DEXP	1-374#	1-374#	1-375	1-375	1-376	1-376	
DIGCNT	1-437#	1-437#					
DIGFLG	1-377#	1-377#					
DIMCNT	1-462#	1-462#					
DIMLP	1-465#	1-465#					
DIMSUB	1-[illegible]#						
DISCNT	1-317#	1-317#					
DL	1-407#	1-407#					
DLTXS	1-280#	1-280#					
DLTYS	1-282#	1-282#					
DN1	1-435#	1-435#					
DN2	1-436#	1-436#					
DOLFLG	1-415#	1-415#					
DOTON	1-315#	1-315#					
DP	1-394#	1-394#					
DPCONT	1-455#	1-455#					
DREXTA	1-185#	1-185#					
DREXTB	1-186#	1-186#					
DSFLG	1-466#	1-466#					
DSPOUT	1-28#						
DSPSTT	1-217#	1-217#					
DSPVCT	1-240#	1-240#					
DT	1-408#	1-408#					
E1	1-427#	1-427#					
E2	1-428#	1-428#					
E3	1-429#	1-429#					
EARLYW	1-329#	1-329#					
EDTBFR	1-133#	1-133#					
EDTEND	1-204#	1-204#					

CROSS REFERENCE TABLE (CREF V01-03)

```
EDTFLG  1-270#  1-270#
EDTMAX  1-206#  1-206#
EDTPTR  1-205#  1-205#
EFLAG   1-379#  1-379#
ENDTBL  1-144#  1-144#
EOFTBL  1-141#  1-141#
EQLCHR  1-231#  1-231#
EOLTG   1-89#   1-89#
EOSTG   1-90#   1-90#
EQU     1-148#  1-148#
ERCTRA  1-172#  1-172#
ERCTRB  1-173#  1-173#
ERRCD   1-169#  1-169#
ERRCDB  1-170#  1-170#
ES      1-426#  1-426#
ESTG    1-67#   1-67#
ETXCHR  1-232#  1-232#
EVLEN   1-22#
EVLSUB  1-22#
EVLTBL  1-22#
EXH     1-375#  1-375#
EXL     1-376#  1-376#
EXP     1-440#  1-440#
EXS     1-441#  1-441#
F1      1-438#  1-438#
F2      1-439#  1-439#
FFLG    1-412#  1-412#
FILES   1-28#
FILFND  1-334#  1-334#
FILLOC  1-333#  1-333#
FLAG    1-150#  1-150#
FLAGS   1-418#  1-418#
FLCTRL  1-22#
FLDATA  1-22#
FMTFLG  1-417#  1-417#
FNTBL   1-142#  1-142#
FONT    1-319#  1-319#
FORTG   1-69#   1-69#
FPA     1-380#  1-380#
FPB     1-381#  1-381#
FPC     1-382#  1-382#
FPMATH  1-19#
FUDGH   1-94#   1-94#
FUDGL   1-95#   1-95#
FULL    1-316#  1-316#
FULSTT  1-321#  1-321#
FUZA    1-383#  1-383#
FUZB    1-384#  1-384#
FUZR    1-385#  1-385#
GETKEY  1-33#
GETLIN  1-20#
GLBFLG  1-182#  1-182#
GOSTG   1-68#   1-68#
GRAF    1-29#
GRAPH   1-32#
HCTRCS  1-136#  1-136#
HEP     1-160#  1-160#
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

```
LDIGR   1-444#  1-444#
LOXX    1-195#  1-195#
LENGTH  1-265#  1-265#
LEX     1-20#
LEXCNT  1-153#  1-153#
LINELN  1-322#  1-322#
LISTTG  1-20#   1-20#
LITFLG  1-409#  1-409#
LNNOTG  1-78#   1-78#
LRSFLG  1-410#  1-410#
LSP     1-165#  1-165#
LSPS    1-442#  1-442#
LSTKEY  1-212#  1-212#
LUNNO   1-237#  1-237#
MAGEND  1-215#  1-215#
MASK    1-155#  1-155#
MATOPR  1-23#
MFLG    1-414#  1-414#
MSKCTR  1-294#  1-294#
MTBFR   1-132#  1-132#
MTDRV   1-30#
MTERR   1-326#  1-326#
MTFLGS  1-331#  1-331#
MTMASK  1-230#  1-230#
MTMAX   1-248#  1-248#
MTPTR   1-249#  1-249#
MTSREG  1-246#  1-246#
MTSTAT  1-247#  1-247#
MTSTT2  1-216#  1-216#
MTXBYT  1-327#  1-327#
MULPNT  1-361#  1-361#
N1      1-430#  1-430#
N11     1-431#  1-431#
N12     1-432#  1-432#
NLPTR   1-178#  1-178#
NOOUT   1-250#  1-250#
NS      1-433#  1-433#
NTPTR   1-179#  1-179#
NULCHR  1-233#  1-233#
NULLTG  1-65#   1-65#
NUMFLG  1-411#  1-411#
ONTBL   1-140#  1-140#
OPRADR  1-183#  1-183#
OPRRTN  1-184#  1-184#
OUTCTL  1-26#
PAETG   1-76#   1-76#
PARCNT  1-405#  1-405#
PARSE   1-19#
PCHAR   1-33#
PFLG    1-413#  1-413#
PGCTR   1-320#  1-320#
PGMEVL  1-22#
PGMLIS  1-20#
PGMPTR  1-159#  1-159#
PGMTG   1-71#   1-71#
PGN     1-61#   1-129   1-129   1-129   1-129#  1-131   1-131   1-131   1-131#  1-132   1-132   1-132   1-132#  1-133
        1-133   1-133   1-133#  1-134   1-134   1-134   1-134#  1-135   1-135   1-135   1-135#  1-140   1-140   1-140
```

CROSS REFERENCE TABLE (CREF V01-03)

	1-140#	1-141	1-141	1-141	1-141#	1-142	1-142	1-142	1-142#	1-143	1-143	1-143	1-143#	1-144
	1-144	1-157	1-157	1-157	1-157#	1-168	1-168	1-168	1-168#	1-171	1-171	1-171	1-171#	1-172
	1-172	1-172	1-172#	1-173	1-173	1-173	1-173#	1-184	1-184	1-184	1-184#	1-188	1-188	1-188
	1-188#	1-189	1-189	1-189	1-189#	1-190	1-190	1-190	1-190#	1-194	1-194	1-194	1-194#	1-195
	1-195	1-195	1-195#	1-196	1-196	1-196	1-196#	1-197	1-197	1-197	1-197#	1-198	1-198	1-198
	1-198#	1-199	1-199	1-199	1-199#	1-209	1-209	1-209	1-209#	1-218	1-218	1-218	1-218#	1-226
	1-226	1-226	1-226#	1-227	1-227	1-227	1-227#	1-228	1-228	1-228	1-228#	1-229	1-229	1-229
	1-229#	1-230	1-230	1-230	1-230#	1-236	1-236	1-236	1-236#	1-240	1-240	1-240	1-240#	1-242
	1-242	1-242	1-242#	1-245	1-245	1-245	1-245#	1-254	1-254	1-254	1-254#	1-270	1-270	1-270
	1-270#	1-272	1-272	1-272	1-272#	1-276	1-276	1-276	1-276#	1-277	1-277	1-277	1-277#	1-278
	1-278	1-278	1-278#	1-279	1-279	1-279	1-279#	1-280	1-280	1-280	1-280#	1-281	1-281	1-281
	1-281#	1-282	1-282	1-282	1-282#	1-283	1-283	1-283	1-283#	1-284	1-284	1-284	1-284#	1-285
	1-285	1-285	1-285#	1-286	1-286	1-286	1-286#	1-287	1-287	1-287	1-287#	1-288	1-288	1-288
	1-288#	1-289	1-289	1-289	1-289#	1-290	1-290	1-290	1-290#	1-291	1-291	1-291	1-291#	1-292
	1-292	1-292	1-292#	1-293	1-293	1-293	1-293#	1-294	1-294	1-294	1-294#	1-295	1-295	1-295
	1-295#	1-296	1-296	1-296	1-296#	1-297	1-297	1-297	1-297#	1-320	1-320	1-320	1-320#	1-321
	1-321	1-321	1-321#	1-322	1-322	1-322	1-322#	1-326	1-326	1-326	1-326#	1-338	1-338	1-338
	1-338#	1-339	1-339	1-339	1-339#	1-340	1-340	1-340	1-340#	1-341	1-341	1-341	1-341#	1-342
	1-342	1-342	1-342#	1-343	1-343	1-343	1-343#	1-344	1-344	1-344	1-344#	1-345	1-345	1-345
	1-345#	1-346	1-346	1-346	1-346#	1-347	1-347	1-347	1-347#	1-348	1-348	1-348	1-348#	1-349
	1-349	1-349	1-349#	1-350	1-350	1-350	1-350#	1-351	1-351	1-351	1-351#	1-377	1-377	1-377
	1-377#	1-378	1-378	1-378	1-378#	1-379	1-379	1-379	1-379#	1-380	1-380	1-380	1-380#	1-381
	1-381	1-381	1-381#	1-382	1-382	1-382	1-382#	1-383	1-383	1-383	1-383#	1-384	1-384	1-384
	1-384#	1-385	1-385	1-385	1-385#	1-386	1-386	1-386	1-386#	1-387	1-387	1-387	1-387#	1-388
	1-388	1-388	1-388#	1-393	1-393	1-393	1-393#	1-395	1-395	1-395	1-395#	1-426	1-426	1-426
	1-426#	1-427	1-427	1-427	1-427#	1-428	1-428	1-428	1-428#	1-429	1-429	1-429	1-429#	1-430
	1-430	1-430	1-430#	1-431	1-431	1-431	1-431#	1-432	1-432	1-432	1-432#	1-433	1-433	1-433
	1-433#	1-434	1-434	1-434	1-434#	1-435	1-435	1-435	1-435#	1-436	1-436	1-436	1-436#	1-437
	1-437	1-437	1-437#	1-438	1-438	1-438	1-438#	1-439	1-439	1-439	1-439#	1-440	1-440	1-440
	1-440#	1-441	1-441	1-441	1-441#	1-442	1-442	1-442	1-442#	1-443	1-443	1-443	1-443#	1-444
	1-444	1-444	1-444#	1-445	1-445	1-445	1-445#	1-446	1-446	1-446	1-446#	1-447	1-447	1-447
	1-447#	1-448	1-448	1-448	1-448#	1-449	1-449	1-449	1-449#	1-450	1-450	1-450	1-450#	1-451
	1-451	1-451	1-451#	1-452	1-452	1-452	1-452#	1-453	1-453	1-453	1-453#	454	1-454	1-454
	1-454#	1-455	1-455	1-455	1-455#	1-456	1-456	1-456	1-456#	1-477	1-477			
PGNEND	1-477#	1-477#												
PGZ	1-60#	1-102	1-102	1-102	1-102#	1-103	1-103	1-103	1-103#	1-104	1-104	1-104	1-104#	1-105
	1-105	1-105	1-105#	1-106	1-106	1-106	1-106#	1-107	1-107	1-107	1-107#	1-108	1-108	1-108
	1-108#	1-109	1-109	1-109	1-109#	1-110	1-110	1-110	1-110#	1-111	1-111	1-111	1-111#	1-112
	1-112	1-112	1-112#	1-113	1-113	1-113	1-113#	1-114	1-114	1-114	1-114#	1-115	1-115	1-115
	1-115#	1-116	1-116	1-116	1-116#	1-117	1-117	1-117	1-117#	1-118	1-118	1-118	1-118#	1-119
	1-119	1-119	1-119#	1-120	1-120	1-120	1-120#	1-121	1-121	1-121	1-121#	1-122	1-122	1-122
	1-122#	1-123	1-123	1-123	1-123#	1-124	1-124	1-124	1-124#	1-125	1-125	1-125	1-125#	1-148
	1-148	1-148	1-148#	1-149	1-149	1-149	1-149#	1-150	1-150	1-150	1-150#	1-151	1-151	1-151
	1-151#	1-152	1-152	1-152	1-152#	1-153	1-153	1-153	1-153#	1-154	1-154	1-154	1-154#	1-155
	1-155	1-155	1-155#	1-156	1-156	1-156	1-156#	1-158	1-158	1-158	1-158#	1-159	1-159	1-159
	1-159#	1-160	1-160	1-160	1-160#	1-161	1-161	1-161	1-161#	1-163	1-163	1-163	1-163#	1-165
	1-165	1-165	1-165#	1-166	1-166	1-166	1-166#	1-167	1-167	1-167	1-167#	1-169	1-169	1-169
	1-169#	1-170	1-170	1-170	1-170#	1-177	1-177	1-177	1-177#	1-178	1-178	1-178	1-178#	1-179
	1-179	1-179	1-179#	1-180	1-180	1-180	1-180#	1-181	1-181	1-181	1-181#	1-182	1-182	1-182
	1-182#	1-183	1-183	1-183	1-183#	1-185	1-185	1-185	1-185#	1-186	1-186	1-186	1-186#	1-203
	1-203	1-203	1-203#	1-204	1-204	1-204	1-204#	1-205	1-205	1-205	1-205#	1-206	1-206	1-206
	1-206#	1-207	1-207	1-207	1-207#	1-208	1-208	1-208	1-208#	1-210	1-210	1-210	1-210#	1-211
	1-211	1-211	1-211#	1-212	1-212	1-212	1-212#	1-213	1-213	1-213	1-213#	1-214	1-214	1-214
	1-214#	1-215	1-215	1-215	1-215#	1-216	1-216	1-216	1-216#	1-217	1-217	1-217	1-217#	1-222
	1-222	1-222	1-222#	1-223	1-223	1-223	1-223#	1-224	1-224	1-224	1-224#	1-225	1-225	1-225
	1-225#	1-231	1-231	1-231	1-231#	1-232	1-232	1-232	1-232#	1-233	1-233	1-233	1-233#	1-234
	1-234	1-234	1-234#	1-235	1-235	1-235	1-235#	1-237	1-237	1-237	1-237#	1-238	1-238	1-238

CROSS REFERENCE TABLE (CREF V01-03)

	1-238#	1-239	1-239	1-239	1-239#	1-246	1-246	1-246	1-246#	1-247	1-247	1-247	1-247#	1-248
	1-248	1-248	1-248#	1-249	1-249	1-249	1-249#	1-250	1-250	1-250	1-250#	1-251	1-251	1-251
	1-251#	1-252	1-252	1-252	1-252#	1-253	1-253	1-253	1-253#	1-255	1-255	1-255	1-255#	1-256
	1-256	1-256	1-256#	1-257	1-257	1-257	1-257#	1-258	1-258	1-258	1-258#	1-259	1-259	1-259
	1-259#	1-260	1-260	1-260	1-260#	1-261	1-261	1-261	1-261#	1-262	1-262	1-262	1-262#	1-263
	1-263	1-263	1-263#	1-265	1-265	1-265	1-265#	1-266	1-266	1-266	1-266#	1-267	1-267	1-267
	1-267#	1-268	1-268	1-268	1-268#	1-269	1-269	1-269	1-269#	1-271	1-271	1-271	1-271#	1-305
	1-305	1-305	1-305#	1-306	1-306	1-306	1-306#	1-307	1-307	1-307	1-307#	1-308	1-308	1-308
	1-308#	1-309	1-309	1-309	1-309#	1-310	1-310	1-310	1-310#	1-311	1-311	1-311	1-311#	1-312
	1-312	1-312	1-312#	1-313	1-313	1-313	1-313#	1-314	1-314	1-314	1-314#	1-315	1-315	1-315
	1-315#	1-316	1-316	1-316	1-316#	1-317	1-317	1-317	1-317#	1-318	1-318	1-318	1-318#	1-319
	1-319	1-319	1-319#	1-327	1-327	1-327	1-327#	1-328	1-328	1-328	1-328#	1-329	1-329	1-329
	1-329#	1-330	1-330	1-330	1-330#	1-331	1-331	1-331	1-331#	1-332	1-332	1-332	1-332#	1-333
	1-333	1-333	1-333#	1-334	1-334	1-334	1-334#	1-355	1-355	1-355	1-355#	1-356	1-356	1-356
	1-356#	1-357	1-357	1-357	1-357#	1-358	1-358	1-358	1-358#	1-359	1-359	1-359	1-359#	1-360
	1-360	1-360	1-360#	1-361	1-361	1-361	1-361#	1-362	1-362	1-362	1-362#	1-363	1-363	1-363
	1-363#	1-364	1-364	1-364	1-364#	1-365	1-365	1-365	1-365#	1-366	1-366	1-366	1-366#	1-367
	1-367	1-367	1-367#	1-368	1-368	1-368	1-368#	1-369	1-369	1-369	1-369#	1-370	1-370	1-370
	1-370#	1-371	1-371	1-371	1-371#	1-372	1-372	1-372	1-372#	1-373	1-373	1-373	1-373#	1-374
	1-374	1-374	1-374#	1-392	1-392	1-392	1-392#	1-394	1-394	1-394	1-394#	1-396	1-396	1-396
	1-396#	1-397	1-397	1-397	1-397#	1-398	1-398	1-398	1-398#	1-399	1-399	1-399	1-399#	1-400
	1-400	1-400	1-400#	1-401	1-401	1-401	1-401#	1-402	1-402	1-402	1-402#	1-403	1-403	1-403
	1-403#	1-404	1-404	1-404	1-404#	1-405	1-405	1-405	1-405#	1-406	1-406	1-406	1-406#	1-407
	1-407	1-407	1-407#	1-408	1-408	1-408	1-408#	1-409	1-409	1-409	1-409#	1-410	1-410	1-410
	1-410#	1-411	1-411	1-411	1-411#	1-412	1-412	1-412	1-412#	1-413	1-413	1-413	1-413#	1-414
	1-414	1-414	1-414#	1-415	1-415	1-415	1-415#	1-416	1-416	1-416	1-416#	1-417	1-417	1-417
	1-417#	1-418	1-418	1-418	1-418#	1-472	1-472	1-472	1-472#	1-476	1-476			
PGZEND	1-476#	1-476#												
PIPEND	1-297#	1-297#												
PIPTBL	1-296#	1-296#												
PLOSTG	1-66#	1-66#												
PNDEOF	1-214#	1-214#												
PNDFLG	1-213#	1-213#												
PNTNTG	1-75#	1-75#												
PNTSTG	1-73#	1-73#												
POINT	1-266#	1-266#												
POINTB	1-254#	1-254#												
PPECF	1-227#	1-227#												
PPEOR	1-226#	1-226#												
PPMODE	1-225#	1-225#												
PPNUL	1-228#	1-228#												
PRINTF	1-25#													
PRTERR	1-19#													
PRTTG	1-87#	1-87#												
PSCTG	1-74#	1-74#												
PWRUP	1-23#													
QEND	1-130#	1-130#												
QIN	1-222#	1-222#												
QOUT	1-223#	1-223#												
R0	1-102#	1-102#	1-419	1-419	1-420	1-420								
R1	1-103#	1-103#	1-421	1-421	1-422	1-422								
R10	1-112#	1-112#												
R11	1-113#	1-113#	1-460	1-460										
R12	1-114#	1-114#	1-461	1-461										
R13	1-115#	1-115#	1-462	1-462										
R14	1-116#	1-116#	1-463	1-463										
R15	1-117#	1-117#	1-464	1-464										

CROSS REFERENCE TABLE (CREF V01-03)

R16	1-118#	1-118#	1-465	1-465		
R17	1-119#	1-119#	1-466	1-466		
R18	1-120#	1-120#	1-467	1-467	1-468	1-468
R19	1-121#	1-121#				
R2	1-104#	1-104#	1-423	1-423	1-424	1-424
R20	1-122#	1-122#				
R21	1-123#	1-123#				
R22	1-124#	1-124#				
R23	1-125#	1-125#				
R3	1-105#	1-105#	1-425	1-425		
R4	1-106#	1-106#				
R5	1-107#	1-107#				
R6	1-108#	1-108#				
R7	1-109#	1-109#				
R8	1-110#	1-110#				
R9	1-111#	1-111#				
RDIG	1-448#	1-448#				
RDIGA	1-449#	1-449#				
RECCNT	1-328#	1-328#				
RECNO	1-238#	1-238#				
RENUMB	1-21#					
RETH	1-467#	1-467#				
RETL	1-468#	1-468#				
ROW	1-342#	1-342#				
ROWCT	1-313#	1-313#				
RTRNTG	1-86#	1-86#				
SBP	1-163#	1-163#				
SCMSED	1-272#	1-272#				
SCORE	1-154#	1-154#				
SCRTCH	1-135#	1-135#	1-136	1-136		
SEMITG	1-82#	1-82#				
SHIFT1	1-340#	1-340#				
SHIFT2	1-341#	1-341#				
SHNTCT	1-151#	1-151#				
SHUNT	1-20#					
SIGNS	1-359#	1-359#				
SINTHA	1-288#	1-288#				
SLOPH	1-97#	1-97#				
SLOPL	1-98#	1-98#				
SP1	1-397#	1-397#				
SP2	1-399#	1-399#				
SP3	1-401#	1-401#				
SP4	1-403#	1-403#				
STAT37	1-234#	1-234#				
STAX	1-198#	1-198#				
STKUSE	1-344#	1-344#				
STPTR	1-158#	1-158#				
STRING	1-25#					
SYMTAB	1-20#					
T1	1-355#	1-355#				
T2	1-356#	1-356#				
T3	1-357#	1-357#				
T4	1-358#	1-358#				
TABCNT	1-456#	1-456#				
TABPNT	1-360#	1-360#				
TABPTR	1-268#	1-268#				
TAGPTR	1-251#	1-251#				

CROSS REFERENCE TABLE (CREF V04-03)

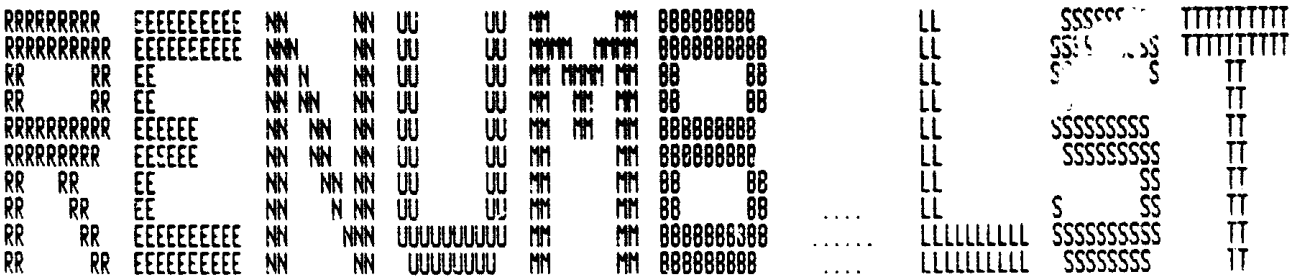
TAPE	1-32#	
TCOL	1-269#	1-269#
TEMPX	1-291#	1-291#
TEMPY	1-293#	1-293#
TFLGS1	1-152#	1-152#
TMP1	1-305#	1-305#
TMP2	1-306#	1-306#
TMP3	1-307#	1-307#
TMPHX	1-310#	1-310#
TMPHY	1-308#	1-308#
TMPLX	1-311#	1-311#
TMPLY	1-309#	1-309#
TPOINT	1-463#	1-463#
TRANSL	1-20#	
TRASH	1-131#	1-131#
TTY2	1-339#	1-339#
TYPIN	1-23#	
UNCOMP	1-21#	
UNLEX	1-21#	
USING	1-24#	
USRORG	1-166#	1-166#
UTILS	1-23#	
VALTG	1-77#	1-77#
VLDKEY	1-351#	1-351#
X0	1-373#	1-373#
X1	1-372#	1-372#
X2	1-371#	1-371#
X3	1-370#	1-370#
X4	1-369#	1-369#
X5	1-368#	1-368#
XAXIS	1-279#	1-279#
XEQSP	1-167#	1-167#
XEQSTK	1-168#	1-168#
XFNBNK	1-236#	1-236#
XFNMAP	1-23#	
XLAST	1-285#	1-285#
XMAXW	1-277#	1-277#
XMINS	1-281#	1-281#
XMINW	1-276#	1-276#
XNEW	1-290#	1-290#
XSF	1-284#	1-284#
Y0	1-367#	1-367#
Y1	1-366#	1-366#
Y2	1-365#	1-365#
Y3	1-364#	1-364#
Y4	1-363#	1-363#
Y5	1-362#	1-362#
YAXIS	1-271#	1-271#
YLAST	1-287#	1-287#
YMAXW	1-279#	1-279#
YMINS	1-283#	1-283#
YMINW	1-278#	1-278#
YNEW	1-292#	1-292#
YSF	1-286#	1-286#
ZERCNT	1-451#	1-451#
ZERO1	1-446#	1-446#
ZERO2	1-447#	1-447#

CROSS REFERENCE TABLE (CREF V01-03)

ZERO3 1-450# 1-450#
ZERSVH 1-452# 1-452#
ZERSVL 1-453# 1-453#
ZX 1-161# 1-161#

CROSS REFERENCE TABLE (CREF V01-03)

ASGN	1-38#	1-65	1-66	1-67	1-68	1-69	1-70	1-71	1-72	1-73	1-74	1-75	1-76	1-77
	1-78	1-79	1-80	1-81	1-82	1-83	1-84	1-85	1-86	1-87	1-88	1-89	1-90	1-94
	1-95	1-97	1-98	1-130	1-136	1-144	1-264	1-375	1-376	1-419	1-420	1-421	1-422	1-423
	1-424	1-425	1-460	1-461	1-462	1-463	1-464	1-465	1-466	1-467	1-468	1-476	1-477	
N	1-51#	1-129	1-131	1-132	1-133	1-134	1-135	1-140	1-141	1-142	1-143	1-157	1-168	1-171
	1-172	1-173	1-184	1-188	1-189	1-190	1-194	1-195	1-196	1-197	1-198	1-199	1-209	1-218
	1-226	1-227	1-228	1-229	1-230	1-236	1-240	1-242	1-245	1-254	1-270	1-272	1-276	1-277
	1-278	1-279	1-280	1-281	1-282	1-283	1-284	1-285	1-286	1-287	1-288	1-289	1-290	1-291
	1-292	1-293	1-294	1-295	1-296	1-297	1-320	1-321	1-322	1-326	1-338	1-339	1-340	1-341
	1-342	1-343	1-344	1-345	1-346	1-347	1-348	1-349	1-350	1-351	1-377	1-378	1-379	1-380
	1-381	1-382	1-383	1-384	1-385	1-386	1-387	1-388	1-393	1-395	1-426	1-427	1-428	1-429
	1-430	1-431	1-432	1-433	1-434	1-435	1-436	1-437	1-438	1-439	1-440	1-441	1-442	1-443
	1-444	1-445	1-446	1-447	1-448	1-449	1-450	1-451	1-452	1-453	1-454	1-455	1-456	
SEI	1-3#													
Z	1-44#	1-102	1-103	1-104	1-105	1-106	1-107	1-108	1-109	1-110	1-111	1-112	1-113	1-114
	1-115	1-116	1-117	1-118	1-119	1-120	1-121	1-122	1-123	1-124	1-125	1-148	1-149	1-150
	1-151	1-152	1-153	1-154	1-155	1-156	1-158	1-159	1-160	1-161	1-163	1-165	1-166	1-167
	1-169	1-170	1-177	1-178	1-179	1-180	1-181	1-182	1-183	1-185	1-186	1-203	1-204	1-205
	1-206	1-207	1-208	1-210	1-211	1-212	1-213	1-214	1-215	1-216	1-217	1-222	1-223	1-224
	1-225	1-231	1-232	1-233	1-234	1-235	1-237	1-238	1-239	1-246	1-247	1-248	1-249	1-250
	1-251	1-252	1-253	1-255	1-256	1-257	1-258	1-259	1-260	1-261	1-262	1-263	1-265	1-266
	1-267	1-268	1-269	1-271	1-305	1-306	1-307	1-308	1-309	1-310	1-311	1-312	1-313	1-314
	1-315	1-316	1-317	1-318	1-319	1-327	1-328	1-329	1-330	1-331	1-332	1-333	1-334	1-355
	1-356	1-357	1-358	1-359	1-360	1-361	1-362	1-363	1-364	1-365	1-366	1-367	1-368	1-369
	1-370	1-371	1-372	1-373	1-374	1-392	1-394	1-396	1-397	1-398	1-399	1-400	1-401	1-402
	1-403	1-404	1-405	1-406	1-407	1-408	1-409	1-410	1-411	1-412	1-413	1-414	1-415	1-416
	1-417	1-418	1-472											


RRRRRRRR EEEEEEEEEE NN NN UU UU MM MM BBBBBBBB LL SSSSSSSS TTTTTTTTTT
RRRRRRRRR EEEEEEEEEE NNN NN UU UU MMMM MMMM BBBBBBBBB LL SSSSSSSSS TTTTTTTTTT
RR RR EE NN N NN UU UU MM MMM MM BB BB LL S S TT
RR RR EE NN NN NN UU UU MM M MM BB BB LL TT
RRRRRRRRR EEEEEE NN NN NN UU UU MM M MM BBBBBBBBB LL SSSSSSSS TT
RRRRRRRR EEEEEE NN NN NN UU UU MM MM BBBBBBBB LL SSSSSSSS TT
RR RR EE NN NN NN UU UU MM MM BB BB LL SS TT
RR RR EE NN N NN UU UU MM MM BB BB .... LL S SS TT
RR RR EEEEEEEEEE NN NNN UUUUUUUUUU MM MM BBBBBBBBB ..... LLLLLLLLLL SSSSSSSSS TT
RR RR EEEEEEEEEE NN NN UUUUUUUU MM MM BBBBBBBB .... LLLLLLLLLL SSSSSSS TT

14-OCT-76

```
                        .TITLE  RENUMB  PROGRAM RENUMBER ROUTINE

                        .GLOBL  RENUMB

                        .IDENT  /JKR009/

                        .GLOBL  PGMPTR
                        .GLOBL  EOLTG,ITM1TG,IMXTG,LNNOTG
                        .GLOBL  ERRCD,ERREN
                        .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
                        .GLOBL  A8X,A9X
                        .GLOBL  FIX1,FLOAT1
                        .GLOBL  PSHFPN,PULFPN
                        .GLOBL  PSHRET,RTRN
                        .GLOBL  SETORG,TYPORG
                        .GLOBL  PUSHX
                        .GLOBL  RENUM
                        .GLOBL  INCXN,LOCIG
                        .GLOBL  GETSMA,GETLAR
                        .GLOBL  DISBLE,ENABLE
                        .GLOBL  IMACOD,ONCOD
                        .GLOBL  GOCOD,PRICOD
                        .GLOBL  LINCOD,CRCOD
```

```
                                        ; RENUMB IS USED TO RENUMBER A PROGRAM POINTED
                                        ; TO BY PGMPTR.  IT ACCEPTS UP TO THREE VALUES ON THE
                                        ; STACK AND SETS DEFAULTS FOR VALUES NOT SEEN AS FOLLOWS:
                                        ;
                                        ;     STARTING # = 100
                                        ;     INCREMENT VALUE = 10
                                        ;     STARTING STATEMENT NUMBER = 100.
                                        ;
                                        ; FOR A PROGRAM THAT HAS NO RENUMBERABLE STATEMENT
                                        ; NUMBERS, RENUMB IS THE SAME AS A NO-OP.
                                        ;
                                        ; RENUMB USES MOST PSEUDO REGISTERS.
                                        ;
                                        ; ERRORS OCCUR IF ANY VALUE IS OUTSIDE, OR IS INCREMENTED
                                        ; OUTISDE, THE RANGE OF 0-65,535; IF THE STARTING
                                        ; NUMBER IS 0; IF THE INCREMENT
                                        ; VALUE = 0; IF THE STARTING PARAMETERS ARE SUCH
                                        ; THAT STATEMENT REPLACEMENT OR INTERLACING WILL OCCUR.
                                        ;
0000  BD  0000G   RENUMB: JSR   PSHRET  ; POP RETURN.
0003  9F    00G           STS   R0,D    ; SET UP R0 FOR PSEUDO STACK PTR.
0005  7F  0000G           CLR   R4      ; CLEAR ARGUMENT FLAGS.
0008  8D    25            BSR   GETVAL  ; TRY TO GET VALUE.
000A  25    14            BCS   SET3    ; NONE THERE.
000C  26    27            BNE   RET     ; ERROR EXIT.
000E  8D    28            BSR   GETVAL  ; TRY TO GET NEXT VALUE.
0010  25    13            BCS   SET2    ; NONE THERE.
0012  26    21            BNE   RET     ; ERROR EXIT.
0014  8D    22            BSR   GETVAL  ; TRY TO GET LAST VALUE.
0016  25    12            BCS   SET1    ; NONE THERE.
0018  26    18            BNE   RET     ; ERROR EXIT.
001A  DE    00G           LDX   R3,D    ; GET LAST VALUE.
001C  DF    00G           STX   R0,D    ; TRANSFER IT FOR RENUM.
001E  20    0F            BRA   SET0    ; NO DEFAULT VALUES.
0020  CE  0064    SET3:   LDX   100.,I  ; DEFAULT LINE # = 100.
0023  DF    00G           STX   R0,D
0025  CE  000A    SET2:   LDX   10.,I   ; DEFAULT INCREMENT = 10.
0028  DF    00G           STX   R1,D
002A  CE  0064    SET1:   LDX   100.,I  ; DEFAULT STATEMENT = 100.
002D  DF    00G           STX   R2,D
002F  DE    00G   SET0:   LDX   PGMPTR,D ; GET PROGRAM POINTER.
]]]]]
0031  27    02            BEQ   RET     ; NO PROGRAM = NO-OP.
0033  8D    39            BSR   RENUM   ; TRY TO DO RENUMBER.
0035  7E  0000G   RET:    JMP   RTRN    ; RETURN.
```

```
                                        ; GETVAL IS USED TO GET VALUES FROM THE STACK AS
                                        ; A RESULT OF A RENUMBER CALL.  IF A VALUE IS
                                        ; PRESENT, IT CONVERTS IT TO AN INTEGER, MOVES R1
                                        ; TO R2, R3 TO R1, AND PLACES THE NEW VALUE
                                        ; IN R3.  IT TAKES AN ERROR EXIT FOR ERRORS.
                                        ; ON EXIT, IF A VALUE WAS SUCCESSFULLY CONVERTED,
                                        ; C = 0; IF NO VALUE IS PRESENT, AN IMMEDIATE
                                        ; RETURN IS MADE WITH C = 1.

0038  BD  0000G     GETVAL: JSR   PSHRET      ; POP RETURN.
003B  DE    0G              LDX   R0,D        ; X POINTS TO STACK.
003D  A6    01              LDA A 1,X         ; GET TAG.
003F  81    0G              CMP A EOLTG,I     ; EOL?
0041  26    07              BNE   BG1         ; NO, CONTINUE.
0043  DE    0G              LDX   R3,D        ; XFER R3 TO R0 FOR RENUM.
0045  DF    0G              STX   R0,D
0047  0D                    SEC               ; SET 'NO ITEM' TAG.
0048  20    E8              BRA   RET         ; RETURN.
004A  BD  0000G     BG1:    JSR   TYPARG      ; TRY TO GET VALUE.
004D  27    06              BEQ   BG2         ; OK VALUE, CONTINUE.
004F  86    0G      BGER:   LDA A ERRENU,I    ; SET ERROR CODE.
0051  97    0G              STA A ERRCD,D
0053  20    16              BRA   CRET        ; ERROR EXIT.
0055  30              BG2:  TSX               ; POINT X AT VALUE.
0056  BD  0000G             JSR   FIX1        ; TRY TO FIX IT.
0059  26    F4              BNE   BGER        ; NOT SUCCESSFUL.
005B  CE    0G              LDX   R1,D        ; BUBBLE UP PREVIOUS VALUES.
005E  DF    0G              STX   R2,D
0060  DE    0G              LDX   R3,D
0062  DF    0G              STX   R1,D
0064  30                    TSX               ; POINT TO FIXED VALUE.
0065  EE    03              LDX   3,X         ; GET NEW VALUE.
0067  DF    0G              STX   R3,D        ; SAVE IT.
0069  9E    0G              LDS   R0,D        ; CLEAR STACK TO PRESENT LEVEL.
006B  4F                    CLR A             ; CLEAR ERROR FLAG.
006C  0C              CRET: CLC               ; C=0=SUCCESSFUL RETURN OR ERROR.
006D  20    C7              BRA   RET         ; RETURN.
```

```
                                        ; RENUM 1  A SPEICAL ENTRY POINT FOR RENUMBERING.
                                        ; ON ENTR
                                        ;
                                        ;       R0 = STARTING NUMBER
                                        ;       R1 = INCREMENT VALUE
                                        ;       R2 = STARTING STATEMENT NUMBER
                                        ;
                                        ; ERROR CONDITIONS ARE SAME AS FOR RENUMB.
                                        ;
                                        ; THE FRONT END PROCESSING IS MAINLY CONCERNED
                                        ; WITH MAKING SURE THE INPUT PARAMETERS IN
                                        ; R0 - R2 MAKE SENSE.
                                        ;
                                        ;
                                        ;
006E  BD  0000G     RENUM:  JSR     PSHRET          ; POP RETURN.
0071  BD  0000G             JSR     DISBLE          ; DISABLE BREAKS.
0074  DE    00G             LDX     R0,D            ; GET STARTING NUMBER.
0076  27    04              BEQ     RENERR          ; CANNOT BE 0, ERROR EXIT.
0078  DE    00G             LDX     R1,D            ; GET INCREMENT VALUE
007A  26    0A              BNE     RC0             ; NO 0, TRY RENUMBER.
007C  86    00G     RENERR: LDA A   ERREN,I         ; GET ERROR CODE.
007E  97    00G             STA A   ERRCD,D         ; FLAG ERROR
0080  BD  0000G     RRET:   JSR     ENABLE          ; ENABLE BREAKS.
0083  7E  0000G             JMP     RTRN            ; RETURN.
0086  DE    00G     RC0:    LDX     R2,D            ; STACK STARTING STATEMENT #.
0088  BD  0000G             JSR     PUSHX
0088  BD  0000G             JSR     GETLAR          ; TRY TO FIND LINE.
008E  30                    TSX                     ; SET UP RESULT.
008F  EE    01              LDX     1,X             ; GET ADDRESS.
0091  26    05              BNE     RC1             ; VALID, CONTINUE.
0093  BD  01AE'             JSR     CLEAN6          ; HAVE NOTHING, CLEAN STACK.
0096  20    E8              BRA     RRET            ; NO-OP RETURN.
0098  DF    00G     RC1:    STX     R4,D            ; SAVE POINTER.
009A  BD  01AE'             JSR     CLEAN6          ; CLEAN STACK.
009D  96    00G             LDA A   R0,D            ; GET STARTING #.
009F  D6    01G             LDA B   R0+1,D
00A1  D0    01G             SUB B   R2+1,D          ; COMPARE TO STARTING STATEMENT.
00A3  92    00G             SBC A   R2,D
00A5  24    24              BCC     RC2             ; 0 = GOING FORWARD.
00A7  DE    00G             LDX     R2,D            ; GET STARTING STATEMENT #.
00A9  09                    DEX                     ; MAKE 1 LESS.
00AA  BD  0000G             JSR     PUSHX           ; STACK IT.
00AD  BD  0000G             JSR     GETSMA          ; TRY TO FIND SMALLER.
00B0  30                    TSX                     ; GET RESULT POINTER.
0081  EE    01              LDX     1,X             ; GET ADDRESS.
0083  26    05              BNE     RCK             ; CHECK FURTHER.
00B5  BD  01AE'             JSR     CLEAN6          ; CLEAN STACK.
00B8  20    2D              BRA     RC3             ; GO FORWARD ONLY.
00BA  EE    07      RCK:    LDX     7,X             ; GET LINE NUMBER.
00BC  DF    00G             STX     R6,D            ; SAVE IT.
00BE  BD  01AE'             JSR     CLEAN6          ; CLEAN STACK.
00C1  96    00G             LDA A   R6,D            ; STARTING # - SMALLER RESULT.
00C3  D6    01G             LDA B   R6+1,D
00C5  D0    01G             SUB B   R0+1,D
00C7  92    00G             SBC A   R0,D
00C9  24    B1              BCC     RENERR          ; RENUMBER A NO-NO FOR THIS SITUATION.
```

```
                                        ; SECTION RC2 REPLACES ALL BACK POINTERS WITH THEIR
                                        ; RESPECTIVE LINE NUMBERS FOR ALL LINES THAT
                                        ; PRECEDE THE FIRST LINE TO BE RENUMBERED.


00CB  8D  30      RC2:  BSR   INCCK     ; INCREMENT OVERFLOW CHECK.
00CD  DE  00G           LDX   R4,D      ; GET STARTING POINT.
00CF  EE  05            LDX   5,X       ; GET BACK POINTER.
00D1  27  16      RLP1: BEQ   RC3S      ; 0 = FINISHED, DO FORWARD.
00D3  DF  00G           STX   R6,D      ; SAVE PRESENT LINE POINTER.
00D5  EE  05            LDX   5,X       ; GET NEXT BACK POINTER.
00D7  DF  00G           STX   R7,D      ; SAVE IT.
00D9  DE  00G           LDX   R6,D      ; GET PRESENT LINE #.
00DB  A6  07            LDA A 7,X       ; MOVE LINE # TO
00DD  A7  05            STA A 5,X       ; BACK POINTER.
00DF  A6  08            LDA A 8,X
00E1  A7  06            STA A 6,X
00E3  DE  00G           LDX   R7,D      ; GET NEXT BACK POINTER.
00E5  20  EA            BRA   RLP1      ; CONTINUE LOOP.
```

```
                                        ; RC3 REPLACES ALL BACK POINTERS WITH THE NEW
                                        ; LINE NUMBER FOR ALL LINES THAT ARE TO BE RENUMBERED.


00E7  8D  14      RC3:   BSR    INCCK     ; INCREMENT OVERFLOW CK.
00E9  DE  00G     RC3S:  LDX    R4,D      ; GET PRESENT LINE POINTER.
00EB  96  00G            LDA A  R0,D      ; GET FIRST LINE #.
00ED  D6  01G            LDA B  R0+1,D
00EF  A7  05      RLP2:  STA A  5,X       ; STORE LINE # IN BACK POINTER.
00F1  E7  06             STA B  6,X
00F3  EE  03             LDX    3,X       ; GET FORWARD POINTER.
00F5  27  1C             BEQ    RC4       ; 0 = FINISHED.
00F7  DB  01G            ADD B  R1+1,D    ; BUMP LINE # BY
00F9  99  00G            ADC A  R1,D      ; INCREMENT VALUE.
00FB  20  F2             BRA    RLP2      ; CONTINUE LOOP.

00FD  DE  00G     INCCK: LDX    R4,D      ; GET STARTING LINE # PTR.
00FF  96  00G            LDA A  R0,D      ; GET STARTING LINE #.
0101  D6  01G            LDA B  R0+1,D
0103  EE  03      RLP3:  LDX    3,X       ; GET FORWARD POINTER.
0105  26  01             BNE    K1        ; ANY LEFT?
0107  39                 RTS              ; NO, SUCCESSFUL RET.
0108  DB  01G     K1:    ADD B  R1+1,D    ; ADD IN INCREMENT.
010A  99  00G            ADC A  R1,D
010C  24  F5             BCC    RLP3      ; NO OVERFLOW, CONTINUE.
010E  32                 PUL A            ; SCRATCH RETURN.
010F  32                 PUL A
0110  7E  007C'          JMP    RENERR    ; ERROR RETURN.
```

```
                                        ; RC4 SEARCHES THE PROGRAM FOR ALL LINES THAT HAVE
                                        ; LINE NUMBER REFERENCES IMBEDDED IN THEM AND
                                        ; CHANGES (UPDATES) ALL LINE NUMBER REFERENCES FOUND
                                        ; TO POINT TO THE NEW LINE NUMBER IF THE
                                        ; CHANGE IS POSSIBLE.

0113  DE  00G       RC4:   LDX   PGMPTR,D     ; INITIALIZE LINE POINTER.
0115  DF  00G       RLP4:  STX   R0,D
0117  A6  09                LDA A  9,X         ; GET LEADING TOKEN.
0119  81  00G               CMP A  IMACOD,I    ; RANGE 1?
011B  25  0E                BCS    RC4A        ; NO, CHECK RANGE 2.
011D  81  00G               CMP A  ONCOD,I     ; YES. = ONCOD?
011F  27  26                BEQ    LCODCK      ; YES, DO LINCOD CHECK.
0121  24  08                BCC    RC4A        ; NO, CHECK RANGE 2.
0123  DE  00G       RLP5:  LDX    R0,D        ; GET NEXT LINE.
0125  EE  03                LDX    3,X
0127  26  EC                BNE    RLP4        ; HAVE ONE, CONTINUE.
0129  20  4E                BRA    RC5         ; NONE LEFT. RESET BACK POINTERS.
012B  A6  00        RC4A:  LDA A  0,X          ; GET LENGTH.
012D  E6  01                LDA B  1,X
012F  DB  01G               ADD B  R0+1,D      ; ADD TO LINE ADDRESS.
0131  99  00G               ADC A  R0,D
0133  97  00G               STA A  R1,D        ; SAVE FOR TRANSFER.
0135  D7  01G               STA B  R1+1,D
0137  DE  00G               LDX    R1,D        ; GET END OF LINE +1 POINTER.
0139  09                   DEX                ; POINT TO LINE TYPE TOKEN.
013A  09                   DEX
013B  A6  00                LDA A  0,X         ; GET LIN TYPE CODE.
013D  81  00G               CMP A  GOCOD,I     ; COMPARE TO - LIMIT.
013F  25  E2                BCS    RLP5        ; NOT IN RANGE
0141  86  00G               LDA A  PRIOR,I     ; GET + LIMIT.
0143  A1  00                CMP A  0,X         ; COMPARE IT.
0145  25  DC                BCS    RLP5        ; OUT OF RANGE, TRY NEXT LINE.
0147  DE  00G       LCODCK: LDX    R0,D        ; GET LINE POINTER.
0149  BD  0000G             JSR    A9X         ; POINT TO FIRST TOKEN.
014C  A6  00        RLP6:  LDA A  0,X          ; GET TOKEN.
014E  81  00G               CMP A  LINCOD,I    ; LINE CODE?
0150  27  09                BEQ    UPDATE      ; YES, UPDATE.
0152  81  00G               CMP A  CRCOD,I     ; END OF LINE.
0154  27  CD                BEQ    RLP5        ; YES, TRY NEXT LINE.
0156  BD  0000G     ULP:   JSR    INCXN       ; POINT TO NEXT TOKEN.
0159  20  F1                BRA    RLP6        ; CONTINUE LOOP.
015B  DF  00G       UPDATE: STX   R6,D        ; SAVE TOKEN POINTER.
015D  EE  01                LDX    1,X         ; GET LINE #.
015F  DF  00G               STX    R7,D        ; SAVE FOR SEARCH.
0161  8D  37                BSR    SEARCH      ; SEARCH FOR LINE #.
0163  25  10                BCS    RTS         ; NOT THERE, LEAVE ALONE.
0165  DE  00G               LDX    R3,D        ; GET LINE POINTER.
0167  EE  05                LDX    5,X         ; GET NEW LINE #.
0169  DF  00G               STX    R7,D        ; SAVE FOR TRANSFER.
016B  DE  00G               LDX    R6,D        ; GET TOKEN POINTER.
016D  96  00G               LDA A  R7,D        ; GET HIGH BYTE.
016F  A7                    STA A  1,X         ; TRANSFER IT TO PROGRAM LINE.
0171  96  01G               LDA A  R7+1,D      ; GET LOW BYTE.
0173  A7  02                STA A  2,X         ; TRANSFER IT TO PROGRAM LINE.
```

```
 58          0175    DE      00G         RTS:    LDX     R6,D            ; POINT TO LAST TOKEN USED.
 59          0177    20      DD                  BRA     ULP             ; CONTINUE TOKEN SEARCH.
```

```
                                        ; RC5 OVERLAYS ALL OLD LINE NUMBERS WITH
                                        ; LINE NUMBERS AND RESTORES THE BACK POINTERS.
                                        ;
                                        ; IT THEN RETURNS TO THE CALLER OF RENUM.
                                        ;
                                        ;
0179  7F  0000G     RC5:  CLR   R0      ; FIRST PREVIOUS LINE
017C  7F  0001G           CLR   R0+1    ; NUMBER = 0.
017F  DE  00G             LDX   PGMPTR,D ; GET STARTING ADDRESS.
0181  A6  05        RLP9: LDA A 5,X     ; TRANSFER LINE NUMBER
0183  A7  07              STA A 7,X     ; FROM BACK POINTER TO
0185  A6  06              LDA A 6,X     ; LINE NUMBER.
0187  A7  08              STA A 8,X
0189  96  00G             LDA A R0,D    ; TRANSFER PREVIOUS
018B  A7  05              STA A 5,X     ; LINE POINTER TO BACK
018D  96  01G             LDA A R0+1,D  ; POINTER.
018F  A7  06              STA A 6,X
0191  DF  00G             STX   R0,D    ; SET UP PREVIOUS LINE POINTER.
0193  EE  03              LDX   3,X     ; GET NEXT LINE POINTER.
0195  26  EA              BNE   RLP9    ; CONTINUE LOOP.
0197  7E  0080'           JMP   RRET    ; FINISHED.
```

```
                                        ; SEARCH IS USED TO SEARCH THE PROGRAM
                                        ; FOR A MATCH WITH THE LINE NUMBER IN R7.
                                        ;
                                        ; FOR AN UNSUCCESSFUL RETURN, C = 1.
                                        ;
                                        ; FOR A SUCCESSFUL RETURN, R3 HAS THE
                                        ; ADDRESS OF THE LINE CONTINING THE LINE
                                        ; NUMBER AND C = 0.
                                        ;
                                        ;
019A  DE  00G   SEARCH: LDX   PGMPTR,0  ; GET STARTING LINE PTR.
019C  DF  00G   RLP7:   STX   R3,0      ; SAVE POINTER.
019E  EE  02            LDX   2,X       ; GET LINE NUMBER.
01A0  9C  00G           CPX   R7,0      ; COMPARE?
01A2  27  08            BEQ   SX        ; YES, SAY SO.
01A4  DE  00G           LDX   R3,0      ; NO. GET PRESENT PTR.
01A6  EE  03            LDX   3,X       ; GET NEXT LINE PTR.
01A8  26  F2            BNE   RLP7      ; GOT ONE, CONTINUE.
01AA  0D                SEC             ; C = 1 FOR FAILURE.
01AB  39                RTS             ; RETURN.
01AC  0C        SX:     CLC             ; C = 0 FOR SUCCESS.
01AD  39                RTS             ; RETURN.
```

```
                                        ; CLEAN6 IS USED TO PRUNE 6 BYTES OFF THE STACK.
                                        ;
                                        ; IT USES R7 AS A WORKING REGISTER.
                                        ;
01AE    30              CLEAN6: TSX                     ; SETUP X.
01AF    EE      00              LDX     0,X             ; GET RETURN ADDRESS.
01B1    DF      00G             STX     R7,0            ; SAVE IT.
01B3    30                      TSX                     ; RESET X.
01B4    BD      0000G           JSR     A8X             ; CLEAN STACK.
01B7    35                      TXS
01B8    DE      00G             LDX     R7,0            ; GET RETURN.
01BA    6E      00              JMP     0,X
                                        ;
                                        ;
        0001'                   .END
```

```
RENUMB   PROGRAM RENUMBER ROUT        RT-11 MMAC  VM02-10   14-OCT-76 01:41:34 PAGE 10+
SYMBOL TABLE

A8X    = xxxxxx G      A9X    = xxxxxx G      BGER     004FR      BG1      004AR      BG2      0055R
CLEAN6   019AER        CRCOD  = xxxxxx G      CRET     006BR      DISBLE= xxxxxx G      ENABLE= xxxxxx G
EOLTG  = xxxxxx G      ERRCD  = xxxxxx G      ERREN  = xxxxxx G      FIX1   = xxxxxx G      FLOAT1= xxxxxx G
GETLAR= xxxxxx G       GETSMA= xxxxxx G       GETVAL   0038R      GOCOD  = xxxxxx G      IMACOD= xxxxxx G
IMXTG  = xxxxxx G      INCCK    00EDR         INCXN  = xxxxxx G      ITM1TG= xxxxxx G      K1       0108R
LCODCK   0147R         LINCOD= xxxxxx G       LNNOTG= xxxxxx G       LOCTG  = xxxxxx G      ONCOD  = xxxxxx G
PGMPTR= xxxxxx G       PRICOD= xxxxxx G       PSHFPN= xxxxxx G       PSHRET= xxxxxx G       PULFPN= xxxxxx G
PUSHX  = xxxxxx G      RCK      00B8R         RC0      0086R      RC1      0098R      RC2      00C8R
RC3      00E7R         RC3S     00E9R         RC4      0113R      RC4A     0128R      RC5      0179R
RENERR   007CR         RENUM    006ERG        RENUMB   0000RG     RET      0035R      RLP1     0001R
RLP2     00EFR         RLP3     0103R         RLP4     0115R      RLP5     0123R      RLP6     014CR
RLP7     019CR         RLP9     0181R         RRET     0080R      RTRN   = xxxxxx G      RTS      0175R
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      SEARCH   019AR      SETARG= xxxxxx G       SET0     002FR
SET1     002AR         SET2     0025R         SET3     0020R      SX       01ACR      TYPARG= xxxxxx G
ULP      0156R         UPDATE   0158R

. ABS.    0000     00
          018C     01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3168. WORDS
,SY:RENUMB/C(DK:1:SETCL!,RENUMB
```

CROSS REFERENCE TABLE (CREF V01-03.)

A8X	1-28#	10-12													
A9X	1-28#	7-37													
BG1	3-16	3-21#													
BG2	3-22	3-26#													
BGER	3-23#	3-28													
CLEANG	4-32	4-35	4-48	4-52	10-8#										
CRCOD	1-40#	7-41													
CRET	3-25	3-38#													
DISBLE	1-37#	4-17													
ENABLE	1-37#	4-24													
EOLTG	1-25#	3-15													
ERRCD	1-26#	3-24*	4-23*												
ERREN	1-26#	3-23	4-22												
FIX1	1-29#	3-27													
FLOAT1	1-29#														
GETLAR	1-36#	4-28													
GETSMA	1-36#	4-44													
GETVAL	2-26	2-29	2-32	3-12#											
GOCOD	1-39#	7-31													
IMACOD	1-38#	7-12													
IMXTG	1-25#														
INCCK	5-7	6-6	6-19#												
INCXN	1-35#	7-43													
ITM1TG	1-25#														
K1	6-23	6-25#													
LCODCK	7-15	7-36#													
LINCOD	1-40#	7-39													
LNNOTG	1-25#														
LOCTG	1-35#														
ONCOD	1-38#	7-14													
PGMPTR	1-24#	2-44	7-9	8-10	9-12										
PRICOD	1-39#	7-33													
PSHFPN	1-30#														
PSHRET	1-31#	2-27	3-12	4-16											
PULFPN	1-30#														
PUSHX	1-33#	4-27	4-43												
R0	1-27#	2-24*	2-36*	2-39*	3-13	3-18*	3-36	4-18	4-36	4-37	4-55	4-56	6-8	6-9	
	6-20	6-21	7-10*	7-17	7-23	7-24	7-36	8-8*	8-9*	8-15	8-17	8-19*			
R1	1-27#	2-41*	3-29	3-32*	4-20	6-14	6-15	6-25	6-26	7-25*	7-26*	7-27			
R10	1-27#														
R11	1-27#														
R2	1-27#	2-43*	3-30*	4-26	4-38	4-39	4-41								
R3	1-27#	2-35	3-17	3-31	3-35*	7-50	9-13*	9-17							
R4	1-27#	2-25*	4-34*	5-8	6-7	6-19									
R5	1-27#														
R6	1-27#	4-51*	4-53	4-54	5-11*	5-14	7-45*	7-53	7-58						
R7	1-27#	5-13*	5-19	7-47*	7-52*	7-54	7-56	9-15	10-10*	10-14					
R8	1-27#														
R9	1-27#														
RC0	4-21	4-26#													
RC1	4-31	4-34#													
RC2	4-40	5-7#													
RC3	4-49	6-6#													
RC3S	5-10	6-7#													
RC4	6-13	7-9#													
RC4A	7-13	7-16	7-21#												

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-03.)

SEL 1-3#

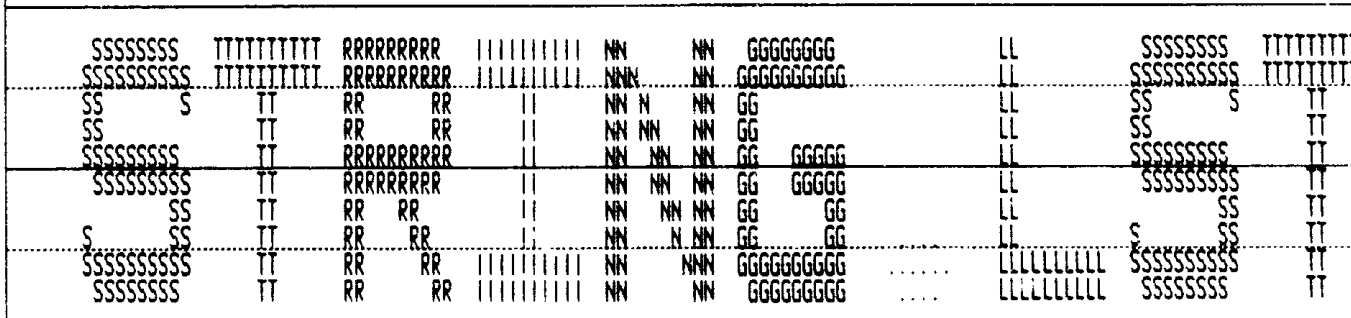


TABLE OF CONTENTS

```
                        .TITLE  STRING  A COLLECTION OF STRING FUNCTIONS
                        .IDENT  /ALL008/
                ;
                        .GLOBL  STRING
0000'           STRING =
                ;
                        .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7 ;PSEUDO REGISTERS
                        .GLOBL  R8,R9,R10,R11
                ;
                        .GLOBL  ERRCD           ;ERROR CODE. NO ERROR=0
                        .GLOBL  ERVAL           ;VAL FUNCTION FAILED
                        .GLOBL  ERSFE           ;STRING FUNCTION ERROR
                        .GLOBL  ERREP           ;REPLACE FUNCTION CAN'T RUN
                        .GLOBL  ERASGN          ;ASSIGN CAN'T WORK
                        .GLOBL  ERFIXN          ;FIX1 ERROR - NEG NUMBER
                ;
                        .GLOBL  AEPLIT, FPALIT, EPC, TESTCS
                        .GLOBL  PSHFPN, PULFPN, FIX1, FLOAT1
                        .GLOBL  PSHRET, RTRN
                        .GLOBL  A5X, A9X, A10X, A14X
                ;
                        .GLOBL  ITM1TG          ;STACK TAG
                ;
                ;       NAME TABLE FORMAT
                ;
0000            NTLINK = 0                      ;POINTER TO NEXT ENTRY
0002            NTNAME = 2                      ;OBJECT NAME
                ;                       FLAG BITS
0004            NTATTR = 4                      ;ATTRIBUTE FLAGS
0080            UNDEF  = H80                    ;UNDEFINED
0040            SCALER = H40                    ;SCALER VALUE
0020            ARRAY  = H20                    ;ARRAY ENTRY
0010            STRING = H10                    ;STRING ENTRY
0008            PARM   = H08                    ;USER FUNCTION PARM
0004            ALLOK  = H04                    ;ALL ELEMENTS OF MATRIX ARE DEFINED
                ;                       VALUE ENTRY
0005            NTVAL  = 5                      ;8 BYTE FLOATING POINT NUMBER
                ;                       ARRAY ENTRY
0005            NTWROW = 5.                     ;WORKING ROW SIZE
0007            NTWCOL = 7.                     ;COLUMN
0009            NTDIMS = 9.                     ;DIMENSIONED SIZE
000B            NTAPTR = 11.                    ;DATA POINTER
                ;                       STRING ENTRY
0005            NTDLEN = 5                      ;DIMENSIONED LENGTH
0007            NTWLEN = 7                      ;WORKING LENGTH
000B            NTSPTR = 11.                    ;DATA POINTER
                ;
                ;       DATA OBJECTS IN MEMORY
                ;
0000            OBJLEN = 0                      ;LENGTH OF ENTRY
0040            VALERR = H40                    ;STRING IS UNDEFINED
0020            OBJSTR = H20                    ;THIS IS A STRING
0002            OBJATR = 2                      ;ATTRIBUTE FLAGS
0003            OBJBCK = 3                      ;BACK POINTER TO NAME TABLE
0005            OBJDT  = 5                      ;DATA
```

```
                                        .SBTTL  *** STRING UTILITIES
                                ;
                                ;       THESE ARE SUB FUNCTIONS FOR ALL THE STRING FUNCTIONS
                                ;
                                ;       SET THE ERROR CODE
                                ;
0000    30              SETERR: TSX                     ;HE JSRED TO ME
0001    EE      00              LDX     0,X             ;AND ERRCD VALUE IS AFTER JSR
0003    A6      00              LDA A   0,X
0005    97      00G             STA A   ERRCD,D
0007    32                      PUL A                   ;FORGET RETURN POINT
0008    32                      PUL A
0009    7E      0000G           JMP     RTRN            ;HE STACKED RETURN ADDR
                                ;
                                ;       SET UP ROUTINE - ABORT PROTECTION COMES FROM SETTING WORKING
                                ;       LENGTH TO ZERO UNTIL DATA IS SAFE.
                                ;       R10 IS USED TO SAVE NT ADDR FOR OUTPUT AREA
                                ;       R11 IS USED TO SAVE NEW WORKING LENGTH
                                ;
000C    DF      00G     STRINT: STX     R10,D           ;HE PASSES NT ADDR IN X
000E    7E      0000G           JMP     PSHRET          ;RESTACK RETURN AND RETURN TO STRING FN
                                ;
                                ;       TERMINATION FUNCTIONS
                                ;
0011    96      00G     STRLAB: LDA A   R11,D           ;GET WORKING LENGTH
0013    D6      01G             LDA B   R11+1,D
0015                    STREND: SEI                     ;BE CAREFULL NOW
0015    01      0F              .BYTE   01,17
0017    DF      00G             LDX     R10,D           ;NT ADDR
0019    A7      07              STA A   NTWLEN,X        ;NEW WORKING LENGTH
001B    E7      08              STA B   NTWLEN+1,X
001D    86      10              LDA A   STRING.I        ;SET UP DEFINED STRING BIT
001F    A7      04              STA A   NTATTR,X
0021    EE      08              LDX     NTSPTR,X        ;GET PTR TO MEMORY
0023    86      20              LDA A   OBJSTR.I        ;MARK IT AS DEFINED
0025    A7      02              STA A   OBJATR,X
0027    0E                      CLI                     ;INTERRUPTS ARE OK NOW
0028    7E      0000G           JMP     RTRN            ;GO TO CALLER
```

```
                                        .SBTTL  XXX VAL FIND VALUE OF ASCII IN STRING
                                        .GLOBL  QVAL
                                ;
                                ;       INPUTS
                                ;               POINTER TO STRING COUNT ON STACK
                                ;
                                ;       OUTPUTS
                                ;               VALUE ON STACK
                                ;
                                ;       NOTES.
                                ;               ERRORS ARE FROM ASCII TO F.P.N.
                                ;
002B  BD  0000G       QVAL:     JSR     PSHRET
002E  30                        TSX                     ;SET UP CONTROL VARS
002F  EE  03                    LDX     3,X             ;POINTER TO STRING COUNT
0031  A6  00                    LDA A   0,X             ;GET COUNT
0033  E6  01                    LDA B   1,X
0035  08                        INX                     ;GET REAL DATA ADD
0036  08                        INX
0037  DF  00G                   STX     R0,D            ;FOR AFPITT
0039  DB  01G                   ADD B   R0+1,D          ;CALC END ADDR
0038  99  00G                   ADC A   R0,D
003D  97  00G                   STA A   R1,D
003F  D7  01G                   STA B   R1+1,D
0041  30                        TSX                     ;PRUNE STACK
0042  BD  0000G                 JSR     A5X
0045  35                        TXS
0046  BD  0000G                 JSR     AFPITT          ;DO CONVERSION
0049  96  00G                   LDA A   ERRCD,D         ;GET ERROR CODE
004B  26  09                    BNE     VALFL           ;IF SET DON'T CHANGE VARIABLE
004D  CE  0000G                 LDX     FPC,I           ;STACK RESULT
0050  BD  0000G                 JSR     PSHFPN
0053  7E  0000G                 JMP     RTRN            ;SO LONG
                                ;
0056  BD  0000'       VALFL:    JSR     SETERR
0059  00G                       .BYTE   ERVAL
```

```
                                        .SBTTL  XXX STR CONVERT NUMBER TO STRING FUNCTION
                                        .GLOBL  QSTR
                                ;
                                ;       INPUTS
                                ;               F.P.N. ON STACK
                                ;               POINTER TO NAME TABLE FOR STRING ENTRY
                                ;
                                ;       OUTPUT
                                ;               ASCII PUT INTO STRING
                                ;
                                ;
005A  30                        QSTR:   TSX                     ;SET UP FOR BREAK
0058  EE    0C                          LDX     12.,X           ;NT ADDR
005D  BD    000C'                       JSR     STRINT
0060  CE    0000G                       LDX     FPC,I           ;PULL OFF INPUT NUMBER
0063  BD    0000G                       JSR     PULFPN
0066  30                                TSX                     ;GET PARMS OFF STACK
0067  EE    01                          LDX     1,X             ;ADDR OF N.T. ENTRY
0069  EE    0B                          LDX     NTSPTR,X        ;ADDR OF DATA IN RAM
006B  BD    0000G                       JSR     A5X
006E  DF    00G                         STX     R1,D
0070  30                                TSX
0071  EE    01                          LDX     1,X
0073  EE    05                          LDX     NTDLEN,X        ;GET MAX SIZE
0075  DF    00G                         STX     R2,D
0077  BD    0000G                       JSR     FPAITT
007A  96    00G                         LDA A   ERRCD,D         ;IF ERROR CODE SET CHANGE IT
007C  26    14                          BNE     SFOOPS          ;USER DOES NOT WANT INTERNAL ERROR CODE
007E  96    00G                         LDA A   R2,D            ;CALC NEW WORKING LEN
0080  D6    01G                         LDA B   R2+1,D
0082  D0    01G                         SUB B   R1+1,D
0084  92    00G                         SBC A   R1,D
0086  CB    01                          ADD B   1,I             ;SO I GET RIGHT ANSWER
0088  89    00                          ADC A   0,I
008A  30                                TSX                     ;NOW PRUNE STRING FROM STACK
008B  BD    0000G                       JSR     A5X
008E  35                                TXS
008F  7E    0015'                       JMP     STREND
                                ;
0092  BD    0000'               SFOOPS: JSR     SETERR
0095  00G                               .BYTE   ERLHSGN         ;DATA WON'T FIT
```

```
                                        .SBTTL  XXX POS STRING SEARCH COMMAND
                                        .GLOBL  QPOS
                                ; FORMAT
                                ;POS(X$,Y$,Z)
                                ;FIND Y$ IN X$ STARTING AT /Z/
                                ;IF Z IS 0, TAKE 1
                                ;IF Z)=0, UPCASE CHARACTORS BEFORE COMPARE
                                ;IF Z<0, DO NOT
                                ;
                                ;REGISTERS
                                ;R0     RETURN ADRESS
                                ;R1     Z+ADRESS OF X$
                                ;R2     TEMP POINTER INTO X$
                                ;R3     ADRESS OF X$+LEN(X$)-LEN(Y$)
                                ;R4     ADRESS OF Y$
                                ;R5     TEMP POINTER INTO Y$
                                ;R6     ADRESS OF Y$+LEN(Y$)
                                ;
0096    BD      0000G   QPOS:   JSR     PSHRET
0099    BD      0000G           JSR     FIX1            ;GET /Z/
009C    81      00G             CMP A   ERFIXN,1        ;NEGITIVE
009E    27      0A              BEQ     2$
00A0    A6      03              LDA A   3,X
00A2    E6      04              LDA B   4,X             ;INTO AB
00A4    C0      01              SUB B   1,1
00A6    82      00              SBC A   0,1             ;MAKE 0 BASED
00A8    24      02              BCC     1$              ;TAKE CARE OF 0 PROBLEM
00AA    4F              2$:     CLR A
00AB    5F                      CLR B
00AC    BD      0000G   1$:     JSR     A9X
00AF    35                      TXS                     ;REMOVE Z FROM STACK
00B0    EB      09              ADD B   9.,X
00B2    A9      08              ADC A   8.,X            ;ADD ADRESS OF X$ TO Z
00B4    25      66              BCS     ZERO            ;WAY TO MANY
00B6    97      00G             STA A   R1,D
00B8    D7      01G             STA B   R1+1,D          ;AND SAVE
00BA    A6      03              LDA A   3,X
00BC    E6      04              LDA B   4,X
00BE    97      00G             STA A   R4,D
00C0    D7      01G             STA B   R4+1,D          ;SAVE ADRESS OF Y$
00C2    97      00G             STA A   R5,D            ;AND SET UP R5
00C4    D7      01G             STA B   R5+1,D
00C6    A6      08              LDA A   8.,X
00C8    E6      09              LDA B   9.,X            ;GET ADRESS OF X$ INTO AB
00CA    EE      08              LDX     8.,X            ;POINT TO LEN OF X$
00CC    EB      01              ADD B   1,X
00CE    A9      00              ADC A   0,X
00D0    30                      TSX
00D1    EE      03              LDX     3,X             ;POINT TO LEN OF Y$
00D3    E0      01              SUB B   1,X
00D5    A2      00              SBC A   0,X             ;FORM INFO FOR R3
00D7    97      00G             STA A   R3,D
00D9    D7      01G             STA B   R3+1,D
                                ;NOW IF THERE IS ANY POSSIBLE COMPARE, R1 IS <=R3
00DB    D0      01G             SUB B   R1+1,D
```

```
 58 00D0 92 00G          SBC A  R1,D
 59 00D2 25 2B           BCS    ZERO          ;NO POSSIBLE TRIES
 60 00E1 A6 CC           LDA A  0,X
 61 00E3 E6 C1           LDA B  1,X           ;LEN OF Y$
 62 00E5 DB 01G          ADD B  R4+1,D
 63 00E7 99 00G          ADC A  R4,D          ;FORM R6 INFO
 64 00E9 97 00G          STA A  R6,D
 65 00EB D7 01G          STA B  R6+1,D
 66 00ED DE 00G          LDX    R1,D
 67 03EF DF 00G    PLP:  STX    R2,D          ;SET TEMP POINTER
 68 00F1 A6 02           LDA A  2,X
 69 00F3 BD 0000G        JSR    TESTCS        ;UPCASE ALL CHARACTERS
 70 00F6 16              TAB                  ;SAVE FOR LATER
 71 00F7 DE 00G          LDX    25,D
 72 00F9 9C 00G          CPX    R6,D          ;CHECK IF ALL COMPARED
 73 00FB 27 24           BEQ    GOTIT         ;THEY DID, SO JOIN
 74 00FD A6 02           LDA A  2,X           ;CHAR OUT OF Y$
 75 00FF 08              INX
 76 0100 DF 00G          STX    R5,D
 77 0102 BD 0000G        JSR    TESTCS
 78 0105 11              CBA                  ;CHECK IF SAME
 79 0106 26 05           BNE    3$            ;TRY SOMEWHERE ELSE
 80 0108 DE 00G          LDX    R2,D          ;STILL OK
 81 010A 08              INX
 82 010B 20 E2           BRA    PLP
 83 010D DE 00G    3$:   LDX    R4,D
 84 010F DF 00G          STX    R5,D          ;RESET POINTERS
 85 0111 DE 00G          LDX    R1,D
 86 0113 9C 00G          CPX    R3,D          ;ANY MORE TRIES LEFT?
 87 0115 27 05           BEQ    ZERO          ;NO
 88 0117 08              INX
 89 0118 DF 00G          STX    R1,D
 90 011A 20 D3           BRA    PLP           ;YES,UPDATE AND TRY AGAIN
 91 011C 4F        ZERO: CLR A
 92 011D 5F              CLR B
 93 011E 30              TSX
 94 011F 20 00           BRA    GOT1
 95 0121 96 00G    GOTIT: LDA A  R1,D
 96 0123 D6 01G          LDA B  R1+1,D        ;COMPUTE Z
 97 0125 30              TSX
 98 0126 E0 09           SUB B  9.,X          ;CHANGE ADRESS BACK TO OFFSET
 99 0128 A2 08           SBC A  8.,X
100 012A CB 01           ADD B  1,I           ;MAKE 1 BASED
101 012C 89 00           ADC A  0,I
102 012E BD 0000G  GOT1: JSR    A10X          ;CLEAN UP STACK
103 0131 35              TXS
104 0132 37              PSH B
105 0133 36              PSH A
106 0134 36              PSH A                ;PHONEY TAG
107 0135 BD 0000G        JSR    FLOAT1        ;FLOAT IT
108 0138 7F 0000G        CLR    ERRCD
109 013B 7E 0000G        JMP    RTRN
```

```
                                        .SBTTL  XXX SEG SEGMENT COMAND
                                        .GLOBL  QSEG
                                ;A$=SEG(X$,Y,Z)
                                ;TAKES /Z/ CHARS OUT OF X$ STARTING AT Y
                                ;IF Z NEG. TAKES THEM IN REVERSE ORDER
                                ;IF RUN OFF END, RETURN NULL STRING

013E  30              QSEG:   TSX
013F  EE  1A                  LDX     26.,X
0141  BD  000C'               JSR     STRINT
0144  BD  0000G               JSR     FIX1            ;FIX Z
0147  E6  03                  LDA B   3,X
0149  D7  00G                 STA B   R11.D
0148  E6  04                  LDA B   4,X             ;INTO A9
014D  D7  01G                 STA B   R11+1.D         ;SAVE Z FOR LATER USE
014F  4D                      TST A                   ;CHECK ERRCD
0150  27  08                  BEQ     3$              ;NOTHING
0152  5F                      CLR B
0153  D7  00G                 STA B   ERRCD.D ;RESET IT
0155  81  00G                 CMP A   ERFIXN,1
0157  26  04                  BNE     3$              ;NOT NEG, WILL TAKE CARE OF SELF
0159  D7  00G                 STA B   R11.D
0158  D7  01G                 STA B   R11+1.D         ;CLEAR, AS INPUT WAS <0
0150  BD  0000G         3$:   JSR     A9X
0160  BD  0000G               JSR     FIX1            ;FIX Y
0163  7F  0000G               CLR     ERRCD
0166  81  00G                 CMP A   ERFIXN,1
0168  27  0A                  BEQ     1$
016A  A6  03                  LDA A   3,X
016C  E6  04                  LDA B   4,X             ;GET Y INTO AB
016E  C0  01                  SUB B   1,1             ;MAKE 0 BASED
0170  82  00                  SBC A   0,1
0172  24  02                  BCC     2$              ;WAS 0, NOW -1, SET BACK TO 0
0174  4F                1$:   CLR A
0175  5F                      CLR B                   ;WAS <=0
0176  97  00G           2$:   STA A   R1.D
0178  D7  01G                 STA B   R1+1.D
017A  BD  0000G               JSR     A9X
017D  35                      TXS                     ;REMOVE Y,Z FROM STACK
017E  DB  01G                 ADD B   R11+1.D
0180  99  00G                 ADC A   R11.D           ;Y+X. OR END OF ANSWER IN INPUT
0182  24  02                  BCC     4$              ;OK STILL
0184  86  FF                  LDA A   377,1           ;FAKE OUT MAX ROUTINE LATTER. Y+Z)2 16
0186  97  00G           4$:   STA A   R2.D
0188  D7  01G                 STA B   R2+1.D
]]]]]
018A  EE  02                  LDX     3,X             ;POINTER TO COUNT OF X$
018C  A6  00                  LDA A   0,X
018E  E6  01                  LDA B   1,X             ;LEN OF X$ IN AB
0190  00                      SEC
0191  D2  01G                 SBC B   R1+1,D          ;CHECK IF Y FALLS IN X$
0193  92  00G                 SBC A   R1.D            ;
0195  24  0A                  BCC     OK              ;YES,FITS
0197  30              NULANS: TSX
0198  BD  0000G               JSR     A10X
0198  35                      TXS
019C  4F                      CLR A
```

```
XXX SEG     SEGMENT COMAND

 58 01901 5F                    CLR B
 59 01902 7E 0015'              JMP     STREND
 60 01905 A6 00         OK:     LDA A   0,X             ;GET LEN OF X$ AGAIN
 61 01907 E6 01                 LDA B   1,X
 62 01909 D1 01G                CMP B   R2+1,D          ;FIND MAX OF LEN X$ AND Y+Z
 63 0190B 92 00G                SBC A   R2,D
 64 0190D 24 0E                 BCC     OK1             ;ALL FIT
 65 0190F A6 00                 LDA A   0,X             ;NO. MUST PRUNE TO LEN OF X$
 66 01911 97 00G                STA A   R2,D
 67 01913 D7 01G                STA B   R2+1,D
 68 01915 D0 01G                SUB B   R1+1,D
 69 01917 92 00G                SBC A   R1,D            ;RESET RESULT LENGTH
 70 01919 97 00G                STA A   R11,D
 71 0191B D7 01G                STA B   R11+1,D
 72 0191D 30            OK1:    TSX
 73 0191E EE 06                 LDX     6,X             ;POINTER TO NT OF RESULT
 74 01920 A6 05                 LDA A   5,X
 75 01922 E6 06                 LDA B   6,X             ;DIMENSIONED LENGTH
 76 01924 D0 01G                SUB B   R11+1,D
 77 01926 82 00G                SBC A   R11,D           ;WILL IT FIT?
 78 01928 24 04                 BCC     OK2             ;YES
 79 0192A BD 0000'      ASNEPR: JSR     SETERR
 80 0192D 00G                   .BYTE   ERASGN
 81 0192E EE 08         OK2:    LDX     11.,X           ;POINTER TO RESULT STORAGE
 82 01930 DF 00G                STX     R3,D
 83 01932 30                    TSX
 84 01933 A6 03                 LDA A   3,X             ;START ADRESS OF X$
 85 01935 E6 04                 LDA B   4,X
 86 01937 DB 01G                ADD B   R1+1,D          ;ADD OFFSETS
 87 01939 99 00G                ADC A   R1,D
 88 0193B 97 00G                STA A   R1,D
 89 0193D D7 01G                STA B   R1+1,D
 90 0193F A6 03                 LDA A   3,X
 91 01941 E6 04                 LDA B   4,X
 92 01943 DB 01G                ADD B   R2+1,D
 93 01945 99 00G                ADC A   R2,D
 94 01947 97 00G                STA A   R2,D
 95 01949 D7 01G                STA B   R2+1,D
 96 0194B BD 0000G              JSR     A10X
 97 0194E 35                    TXS                     ;REMOVE EVERYTHING FROM STACK
 98 0194F DE 00G        2$      LDX     R1,D            ;DO ACTUAL MOVE
 99 01951 9C 00G                CPX     R2,D
100 01953 27 0E                 BEQ     RETSG
101 01955 A6 02                 LDA A   2,X             ;2 IS FROM COUNT SPACE
102 01957 08                    INX
103 01958 DF 00G                STX     R1,D
104 0195A DE 00G                LDX     R3,D            ;FROM HERE TO THERE
105 0195C A7 05                 STA A   5,X
106 0195E 08                    INX
107 0195F DF 00G                STX     R3,D
108 01961 20 EC                 BRA     2$
109
110 01963 7E 0011'      RETSG:  JMP     STRLAB
```

```
                                        .SBTTL  XXX CAT CONCATANATE COMAND
                                        .GLOBL  QCAT
                                ;FORM A$=X$&Y$
                                ;
                                ;MUST BE DONE IN REVERSE ORDER
                                ;PUT Y$IN FIRST FORM THE REAR, THEN X$
                                ;TAKES CARE OF ALL CASES,I THINK
                                ;
0202  30              QCAT:   TSX
0203  EE  00                  LDX     13.,X
0205  BD  000C'              JSR     STRINT
0208  EE  03                  LDX     3,X
020A  A6  00                  LDA A   0,X             ;LEN OF Y$
020C  E6  01                  LDA B   1,X
020E  30                      TSX
020F  EE  08                  LDX     8.,X
0211  EB  01                  ADD B   1,X             ;ADD LEN OF X$
0213  A9  00                  ADC A   0,X
0215  30                      TSX
0216  36                      PSH A                   ;TEMP SAVE
0217  EE  08                  LDX     11.,X           ;PNTR TO MT OF A$
0219  E1  06                  CMP B   6,X             ;CHECK IF ENOUGH ROOM
021B  A2  05                  SBC A   5,X
021D  25  09                  BCS     COK             ;FITS
021F  26  04                  BNE     1$              ;DEFINATLY DOSNT
0221  E1  06                  CMP B   6,X             ;RECHECK B
0223  27  03                  BEQ     COK             ;JUST FITS
0225  32            1$:     PUL A                   ;REMOVE THAT GARBAGE
0226  20  9E                  BRA     ASNERR
0228  32            COK:    PUL A                   ;RESTORE LENGTH
0229  97  00G                 STA A   R11.D           ;SET CURR LENGTH
022B  D7  01G                 STA B   R11+1.D
022D  EB  0C                  ADD B   12.,X           ;ADD START ADRESS
022F  A9  08                  ADC A   11.,X
0231  97  00G                 STA A   R1.D            ;OUTPUT POINTER
0233  D7  01G                 STA B   R1+1.D
0235  4F                      CLR A
0236  97  00G                 STA A   24.D            ;FLAG FOR PASS 1 OR 2
0238  30            LOOP:   TSX
0239  A6  03                  LDA A   3,X             ;GET ADR OF Y$(OR X$)
023B  E6  04                  LDA B   4,X
023D  97  00G                 STA A   R2.D            ;AND SAVE
023F  D7  01G                 STA B   R2+1.D
0241  BD  0000G               JSR     A5X
0244  35                      TXS                     ;REMOVE ARG FROM STACK
0245  DE  00G                 LDX     R2.D
0247  EB  01                  ADD B   1,X             ;ADD LENGTH TO START
0249  A9  00                  ADC A   0,X
024B  97  00G                 STA A   R3.D            ;RIGHT END OF INPUT
024D  D7  01G                 STA B   R3+1.D
024F  DE  00G       LP1:      LDX     R3.D            ;IN POINTER
0251  9C  00G                 CPX     R2.D
0253  27  0E                  BEQ     1$              ;AT START SO DONE
0255  09                      DEX
0256  A6  02                  LDA A   2,X             ;GET CHARACTOR
0258  DF  00G                 STX     R3.D
025A  DE  00G                 LDX     R1.D            ;OUT POINTER
```

```
 58  025C  09                    DEX
 59  025D  A7  05                STA A  5,X
 60  025F  DF  00G               STX    R1,D
 61  0261  20  EC                BRA    LP1
 62  0263  B6  0000G     1$:     LDA A  R4          ;CHECK PASS
 63  0266  26  05                BNE    2$          ;DONE
 64  0268  7C  0000G             INC    R4          ;SET PASS 2
 65  026B  20  CB                BRA    LOOP        ;AND DO IT
 66  026D  30                2$: TSX
 67  026E  BD  0000G             JSR    A5X         ;PRUNE OFF NT INFO
 68  0271  35                    TXS
 69  0272  7E  0011'             JMP    STRLAB
```

```
                                        .SBTTL  XXX REP LACE FUNCTION
                                        .GLOBL  QREP
                                ;A$=REP(X$,Y,Z) REPALCES Z CHARS AT Y IN A$ WITH X$
                                ;REGISTERS USED:
                                ; R0    RETURN ADRESS
                                ; R1    Y+/X$/
                                ; R2    /A$/+/X$/-Z
                                ; R3    Y+Z
                                ; R4    /A$/
                                ; R5    Y
                                ; R6    POINTER TO X$
                                ;
                                QREP:   TSX
0275    30
0276    EE      1A              LDX     26.,X
0278    BD      000C'           JSR     STRINT
027B    BD      0000G           JSR     FIX1            ;FIX Z
027E    26      78              BNE     CRPOUT          ;DIDN'T LIKE Z
0280    A6      03              LDA A   3,X
0282    2B      74              BMI     CRPOUT          ;MUCH TO BIG
0284    E6      04              LDA B   4,X
0286    97      00G             STA A   R2,D
0288    D7      01G             STA B   R2+1,D          ;PUT INTO R2
028A    BD      0000G           JSR     A9X
028D    BD      0000G           JSR     FIX1            ;FIX Y
0290    26      66              BNE     CRPOUT          ;DON'T TAKE ANY NONSENSE
0292    A6      03              LDA A   3,X
0294    2B      62              BMI     CRPOUT
0296    E6      04              LDA B   4,X             ;GET Y INTO ACC5
0298    C0      01              SUB B   1,I
029A    82      00              SBC A   0,I             ;MAKE 0 BASED
029C    25      5A              BCS     CRPOUT          ;WAS 0, NO GOOD
029E    97      00G             STA A   R5,D
02A0    D7      01G             STA B   R5+1,D          ;SET UP R5
02A2    DB      01G             ADD B   R2+1,D
02A4    99      00G             ADC A   R2,D            ;X+Y
02A6    97      00G             STA A   R3,D
02A8    D7      01G             STA B   R3+1,D
02AA    BD      0000G           JSR     A9X
02AD    35                      TXS                     ;PRUNE Y,Z OFF STACK
02AE    EE      03              LDX     3,X             ;POINTER TO LEN OF X$
02B0    A6      00              LDA A   0,X
02B2    E6      01              LDA B   1,X             ;LEN
02B4    DB      01G             ADD B   R5+1,D
02B6    99      00G             ADC A   R5,D            ;PLUS Y
02B8    97      00G             STA A   R1,D
02BA    D7      01G             STA B   R1+1,D
02BC    A6      00              LDA A   0,X
]]]]]
02BE    E6      01              LDA B   1,X
02C0    D0      01G             SUB B   R2+1,D
02C2    92      00G             SBC A   R2,D            ;/X$/-Z
02C4    97      00G             STA A   R2,D
02C6    D7      01G             STA B   R2+1,D
02C8    97      00G             STA A   R7,D            ;SAVE, AS SIGN IS OF INTEREST
02CA    09                      DEX
02CB    09                      DEX
02CC    09                      DEX                     ;MAKE OFF BY 5, SAME AS EVERYONE ELSE
```

```
 58  02CD  DF    00G             STX     R6,D
 59  02CF  30                    TSX
 60  02D0  BD    0000G           JSR     A5X
 61  02D3  35                    TXS                     ;PRUNE X$ OFF
 62  02D4  EE    01              LDX     1,X             ;  TO NT OF A$
 63  02D6  A6    07              LDA A   7,X
 64  02D8  E6    08              LDA B   8.,X            ;LEN OF A$
 65  02DA  D1    01G             CMP B   R3+1,D          ;SEE IF ALL DELETE STUFF LANDS IN STRING
 66  02DC  92    00G             SBC A   R3,D
 67  02DE  25    18              BCS     CRPOUT
 68  02E0  A6    07              LDA A   7,X
 69  02E2  97    00G             STA A   R4,D
 70  02E4  D7    01G             STA B   R4+1,D
 71  02E6  DB    01G             ADD B   R2+1,D
 72  02E8  99    00G             ADC A   R2,D            ;/A$/+/X$/-Z
 73  02EA  97    00G             STA A   R2,D
 74  02EC  D7    01G             STA B   R2+1,D
 75  02EE  A6    05              LDA A   5,X
 76  02F0  E6    06              LDA B   6,X             ;DIM LEN OF RESULT
 77  02F2  D0    01G             SUB B   R2+1,D
 78  02F4  92    00G             SBC A   R2,D            ;SEE IF IT WILL FIT
 79  02F6  24    04              BCC     OKGO
 80  02F8  BD    0000'   CRPOUT: JSR     SETERR
 81  02FB  00G                   .BYTE   ERREP
 82  02FC  96    00G     OKGO:   LDA A   R2,D
 83  02FE  D6    01G             LDA B   R2+1,D          ;SET NEW LEN
 84  0300  97    00G             STA A   R11,D
 85  0302  D7    01G             STA B   R11+1,D
 86  0304  EE    08              LDX     11.,X           ;PNT TO STORAGE
 87  0306  DF    00G             STX     R0,D            ;SAVE ,THEN ADD TO ALL R1-R5
 88  0308  CE    0000G           LDX     R1,I
 89  030B  A6    00      1$:     LDA A   0,X
 90  030D  E6    01              LDA B   1,X
 91  030F  D8    01G             ADD B   R8+1,D
 92  0311  99    00G             ADC A   R8,D
 93  0313  A7    00              STA A   0,X
 94  0315  E7    01              STA B   1,X
 95  0317  08                    INX
 96  0318  08                    INX
 97  0319  9C    0000G           CPX     R6,I
 98  031C  26    ED              BNE     1$              ;NOW THEY ARE ALL ADRESSES INTO A$
 99  031E  96    00G     MOVIT:  LDA A   R7,D
100  0320  2B    14              BMI     10$             ;MOV TO LEFT
101  0322  DE    00G     1$:     LDX     R4,D            ;RIGHT
102  0324  9C    00G             CPX     R3,D
103  0326  27    2$              BEQ     STUFF           ;DONE MOVING
104  0328  09                    DEX
105  0329  DF    00G             STX     R4,D
106  032B  A6    05              LDA A   5,X
107  032D  DE    00G             LDX     R2,D
108  032F  09                    DEX
109  0330  A7    05              STA A   5,X             ;ACTUAL CHARACTOR MOVE
110  0332  DF    00G             STX     R2,D
111  0334  20    EC              BRA     1$
112  0336  DE    00G     10$:    LDX     R1,D            ;MOVE TO LEFT, SAVE R1
113  0338  DF    00G             STX     R2,D
114  033A  DE    00G     2$:     LDX     R3,D
```

```
115  033C  9C  00G          CPX   R4,D
116  033E  27  0E           BEQ   STUFF     ;DONE WITH MOVE
117  0340  A6  05           LDA A 5,X
118  0342  08               INX
119  0343  DF  00G          STX   R3,D
120  0345  DE  00G          LDX   R7,D
121  0347  A7  05           STA A 5,X
122  0349  08               INX
123  034A  DF  00G          STX   R7,D
124  034C  20  EC           BRA   2$        ;MORE TO MOVE
125  034E  DE  00G  STUFF:  LDX   R6,D      ;MOVE SOURCE DATA IN NOW
126  0350  9C  00G          CPX   R3,D
127  0352  26  02           BNE   1$        ;TAKE CARE OF SOURCE AND DEST SAME
128  0354  DE  00G          LDX   R1,D
129  0356  A6  05     1$:   LDA A 5,X       ;GET CHAR FROM X$
130  0358  08               INX
131  0359  DF  00G          STX   R6,D
132  035B  DE  00G          LDX   R5,D      ;DEST
133  035D  9C  00G          CPX   R1,D
134  035F  27  07           BEQ   NOMORE    ;ALL DONE
135  0361  A7  05           STA A 5,X
136  0363  08               INX
137  0364  DF  00G          STX   R5,D
138  0366  20  E6           BRA   STUFF
139  0368  30         NOMORE: TSX
140  0369  BD  0000G        JSR   A$X       ;REMOVE A$ FROM STACK
141  036C  35               TXS
142  036D  7E  0011'        JMP   STRLAB
```

```
                                        .SBTTL  XXX LEN LENGTH OF STRING FUNCTION
                                        .GLOBL  QLEN            ; LENGTH OF STRING
                                        ;
                                        ; LEN IS THE LENGTH FUNCTION HANDLER.  IT CHANGES
                                        ; THE PSCTG STACK ENTRY INTO A VALTG ENTRY WHICH
                                        ; REPRESENTS THE LENGTH (BYTE COUNT) OF THE STRING
                                        ; POINTED TO BY THE PSCTG.  LEN CAN NEVER
                                        ; GENERATE AN ERROR.  ASSIGNMENT IS DONE BY THE EVALUATOR
                                        ;
                                        ;
0370  BD  0000G       QLEN:  JSR     PSHRET          ; POP RETURN.
0373  30                    TSX                     ; SET UP INDES.
0374  EE  03                LDX     3,X             ; GET BYTE COUNT POINTER.
0376  32                    PUL A                   ; CLEAN STACK.
0377  32                    PUL A
0378  32                    PUL A
0379  32                    PUL A
037A  32                    PUL A
037B  A6  00                LDA A   0,X             ; GET BYTE COUNT.
037D  E6  01                LDA B   1,X
037F  37              STGFIN: PSH B                   ; STACK IT.
0380  36                    PSH A
0381  36                    PSH A
0382  BD  0000G             JSR     FLOAT1          ; MAKE FLOATING POINT.
0385  7E  0000G       STGRET: JMP     RTRN            ; RETURN
```

```
                                        .SBTTL  *** ASC ASCII TO FPN FUNCTION
                                        .GLOBL  QASC            ; CHARACTER TO VALUE
                                        ;
                                        ; ASC IS THE CHARACTER-TO-VALUE STRING FUNCTION
                                        ; HANDLER.  IT CONVERTS THE CHARACTER POINTED TO
                                        ; BY THE PSCTG ENTRY INTO A FP VALUE ON
                                        ; THE STACK AS A VALTG ENTRY.  AN ERROR OCCURS
                                        ; IF THE STRING BYTE COUNT IS 0.
                                        ; ASSIGNMENT IS DONE BY THE EVALUATOR.
                                        ;
                                        ;
0388  BD  0000G       QASC:   JSR     ?SHRET          ; POP  RETURN.
0388  30                      TSX                     ; SET UP POINTER
038C  EE  03                  LDX     3,X             ; GET DATA POINTER.
038E  DF  00G                 STX     R0,D            ; SAVE IT.
0390  32                      PUL A                   ; CLEAN STACK
0391  32                      PUL A
0392  32                      PUL A
0393  32                      PUL A
0394  32                      PUL A
0395  EE  00                  LDX     0,X             ; GET BYTE COUNT.
0397  27  1A                  BEQ     CER             ; NO CHARACTERS.  ERROR EXIT
0399  DE  00G                 LDX     R0,D            ; YES, RESTORE DATA POINTER
0398  E6  02                  LDA B   2,X             ; GET CHARACTER VALUE.
039D  4F                      CLR A                   ; CLEAR HIGH ORDER
039E  20  DF                  BRA     STGFIN          ; FINISH.
```

```
                                        .SBTTL  XXX CRH FPN TO ASCII FUNCTION
                                        .GLOBL  QCHR            ; VALUE TO CHARACTER

                                        ; CHR IS THE VALUE-TO-CHARACTER STRING FUNCTION
                                        ; HANDLER.  IT CONVERTS THE VALTG ENTRY ON THE STACK
                                        ; TO AN INTEGER AND ASSIGNS THE RESULTANT VALUE
                                        ; TO THE STRING VARIABLE POINTED TO BY THE PNTSTG
                                        ; STACK ENTRY.  AN ERROR RESULTS IF THE VALTG VALUE
                                        ; IS NEGATIVE OR GREATER THAN 127.


03A0    30              QCHR:   TSX
03A1    EE      0C              LDX     12.,X
03A3    BD      000C'           JSR     STRINT
03A6    30                      TSX                     ; SET UP INDEX FOR FIX1.
03A7    BD      0000G           JSR     FIX1            ; CONVERT FP TO INTEGER.
03AA    96      00G             LDA A   ERRCD,D         ; ANY ERRORS?
03AC    26      05              BNE     CER             ; YES, FLAG IT.
03AE    30                      TSX                     ; NO. GET VALUE POINTER.
03AF    A6      03              LDA A   3,X             ; GET HIGH ORDER BYTE.
0381    27      04              BEQ     B2              ; 0=OK.
03B3    BD      0000'   .       CER:    JSR     SETERR
03B6    00G                     .BYTE   ERSFE
03B7    A6      04      B2:     LDA A   4,X             ; GET LOW ORDER BYTE.
03B9    85      80              BIT A   200.1           ; MUST BE 127 OR LESS
03BB    26      F6              BNE     CER             ; BIT 7 SET, TOO BIG.
03BD    EE      0A              LDX     10.,X           ; GET NAME TABLE ADDRESS
03BF    EE      0B              LDX     11.,X           ; GET DATA OBJECT ADDR
03C1    A7      05              STA A   5,X             ; PUT ASCII DATA INTO STRING
03C3    30                      TSX                     ; PRUNE STACK
03C4    BD      0000G           JSR     A14X
03C7    35                      TXS
03C8    4F                      CLR A                   ; SET NEW LENGTH TO ONE
03C9    C6      01              LDA B   1.1
03CB    7E      0015'           JMP     STREND
                        ;
        0001'                   .END
```

```
STRING A COLLECTION OF STRING  RT-11 MMAC  VM02-10  14-OCT-76 01:41:44 PAGE 11+
SYMBOL TABLE

AFPITT= XXXXXX G    ALLOK =  0004      ARRAY =  0020      ASNERR  01C6R     A10X  = XXXXXX G
A14X  = XXXXXX G    A6X   = XXXXXX G   A9X   = XXXXXX G   B2      0387R     CER     0383R
COK     0228R       CRPOUT  02F8R     ERASGN= XXXXXX G   ERFIXN= XXXXXX G   ERRCD = XXXXXX G
EREP  = XXXXXX G    ERSFE = XXXXXX G   ERVAL = XXXXXX G   FIX1  = XXXXXX G   FLOAT1= XXXXXX G
FPAITT= XXXXXX G    FPC   = XXXXXX G   GOTIT   0121R     GOT1    0312ER    ITM1TG= XXXXXX G
LOOP    0238R       LP1     024FR      MOVIT   031ER      NOMORE  0368R     NTAPTR=  0008
NTATTR=  0004       NTDIMS=  0009      NTDLEN=  0005      NTLINK=  0000     NTNAME=  0002
NTSPTR=  0008       NTVAL =  0006      NTWCOL=  0007      NTWLEN=  0007     NTWROW=  0005
NULANS  0197R       OBJATR=  0002      OBJBCK=  0003      OBJCT =  0005     OBJLEN=  0000
OBJSTR=  0020       OK      01A1R      OKGO    02FCR      OK1     0189R     OK2     01CAR
PARM  =  0008       PLP     00EFR      PSHFPN= XXXXXX G   PSHRET= XXXXXX G   PULFPN= XXXXXX G
QASC    0388RG      QCAT    0202RG     QCHR    03A0RG     QLEN    0370RG    QPOS    0096RG
QREP    0275RG      QSEG    013ERG     QSTR    005ARG     QVAL    002BRG    RETSG   01FFR
RTRN  = XXXXXX G    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   SCALER=  0040     SETERR  0000R
SEOPS   0092R       STGFIN  037FR      STGRET  0385R      STREND  0016R     STRING=  0010 G
STRINT  000CR       STRLAB  0011R      STUFF   034ER      TESTCS= XXXXXX G   UNDEF =  0080
VALERR=  0040       VALFL   0056R      ZERO    011CR
  ABS.    0000      00
          03CE      01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3043. WORDS
.SY:STRING/C<DK1:SEICLI.STRING
```

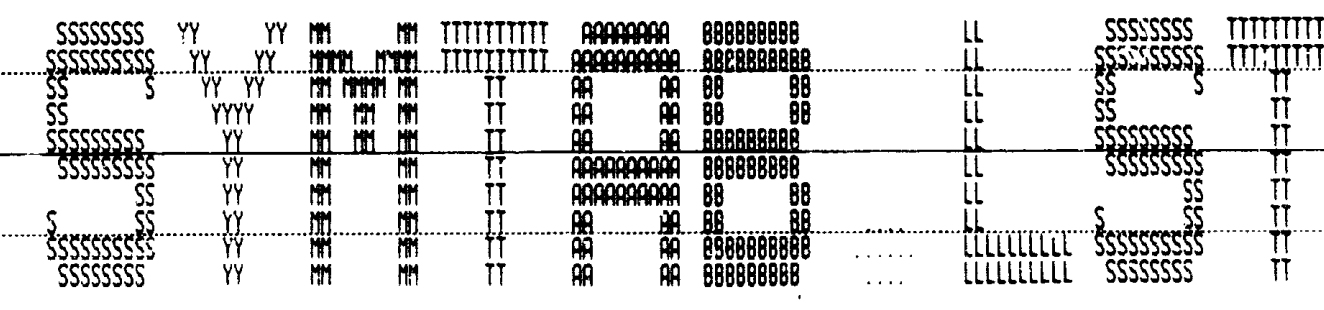
CROSS REFERENCE TABLE (CREF VG1-03)

	1-20							
A10X	1-35#	5-102	6-55	6-96				
A14X	1-35#	11-32						
A5X	1-35#	3-26	4-20	4-36	7-44	7-67	8-60	8-140
A9X	1-35#	5-32	6-25	6-39	8-24	8-39		
AFPITT	1-32#	3-28						
ALLOK	1-50#							
ARRAY	1-47#							
ASNERR	6-79#	7-29						
B2	11-22	11-25#						
CER	10-23	11-19	11-23#	11-27				
COK	7-24	7-27	7-30#					
CRPOUT	8-18	8-20	8-26	8-28	8-32	8-67	8-80#	
ERASGN	1-29#	4-41	6-80					
ERFIXN	1-30#	5-23	6-21	6-28				
ERRCD	1-25#	2-10*	3-29	4-27	5-108*	6-20*	6-27*	11-18
ERREP	1-28#	8-81						
ERSFE	1-27#	11-24						
ERVAL	1-26#	3-36						
FIX1	1-33#	5-22	6-12	6-26	8-17	8-25	11-17	
FLOAT1	1-33#	5-107	9-25					
FPAITT	1-32#	4-26						
FPC	1-32#	3-31	4-15					
GOT1	5-94	5-102#						
GOTIT	5-73	5-95#						
ITM1TG	1-37#							
LOOP	7-39#	7-65						
LP1	7-51#	7-61						
MOVIT	8-99#							
NOMORE	8-134	8-139#						
NTAPTR	1-57#							
NTATTR	1-44#	2-32*						
NTDIMS	1-56#							
NTDLEN	1-59#	4-24						
NTLINK	1-41#							
NTNAME	1-42#							
NTSPTR	1-61#	2-33	4-19					
NTVAL	1-52#							
NTWCOL	1-55#							
NTWLEN	1-60#	2-29*	2-30*					
NTWROW	1-54#							
NULANS	6-54#							
OBJATR	1-68#	2-35*						
OBJBCK	1-69#							
OBJDT	1-70#							
OBJLEN	1-65#							
OBJSTR	1-67#	2-34						
OK	6-53	6-60#						
OK1	6-64	6-72#						
OK2	6-78	6-81#						
OKGO	8-79	8-82#						
PARM	1-49#							
PLP	5-67#	5-82	5-90					
PSHFPN	1-33#	3-32						
PSHRET	1-34#	2-21	3-13	5-21	9-12	10-13		
PULFPN	1-33#	4-16						

CROSS REFERENCE TABLE (CREF V01-03)

QBCC	10-2#	10-13#												
QCNT	7-2#	7-9#												
QCHR	11-2#	11-13#												
QLEN	9-2#	9-12#												
QPOS	5-2#	5-21#												
QREP	8-2#	8-14#												
QSEG	6-2#	6-9#												
QSTR	4-2#	4-12#												
QVAL	3-2#	3-13#												
R0	1-22#	3-20#	3-21	3-22	10-16#	10-24								
R1	1-22#	3-23#	3-24#	4-21#	4-31	4-32	5-37#	5-38#	5-57	5-58	5-66	5-85	5-89#	5-95
	5-96	6-37#	6-38#	6-51	6-52	6-68	6-69	6-86	6-87	6-88#	6-89#	6-98	6-103#	7-35#
	7-36#	7-57	7-60#	8-46#	8-47#	8-88	8-112	8-128	8-133					
R10	1-23#	2-20#	2-28											
R11	1-23#	2-25	2-26	6-14#	6-16#	6-23#	6-24#	6-41	6-42	6-70#	6-71#	6-76	6-77	7-31#
	7-32#	8-84#	8-85#											
R2	1-22#	4-25#	4-29	4-30	5-67#	5-80	6-45#	6-46#	6-62	6-63	6-66#	6-67#	6-92	6-93
	6-94#	6-95#	6-99	7-42#	7-43#	7-46	7-52	8-22#	8-23#	8-35	8-36	8-50	8-51	8-52#
	8-53#	8-71	8-72	8-73#	8-74#	8-77	8-78	8-82	8-83	8-107	8-110#			
R3	1-22#	5-54#	5-55#	5-86	6-82#	6-104	6-107#	7-49#	7-50#	7-51	7-56#	8-37#	8-38#	8-65
	8-66	8-102	8-114	8-119#	8-126									
R4	1-22#	5-41#	5-42#	5-62	5-63	5-83	2-38#	7-62	7-64#	8-69#	8-70#	8-101	8-105#	8-115
R5	1-22#	5-43#	5-44#	5-71	5-76#	5-84#	8-33#	8-34#	8-44	8-45	8-132	8-137#		
R6	1-22#	5-64#	5-65#	5-72	8-58#	8-97	8-125	8-131#						
R7	1-22#	8-54#	8-99	8-112#	8-120	8-123#								
R8	1-23#	8-87#	8-91	8-92										
R9	1-23#													
RETSG	6-100	6-110#												
RTRN	1-34#	2-13	2-37	3-33	5-109	9-26								
SCALER	1-46#													
SETERR	2-7#	3-35	4-40	6-79	8-80	11-23								
SFOOPS	4-28	4-40#												
STGFIN	9-22#	10-27												
STGRET	9-26#													
STREND	2-27#	4-38	6-59	11-36										
STRING	1-19#	1-20#	1-48#	2-31										
STRINT	2-20#	4-14	6-11	7-11	8-16	11-15								
STRLAB	2-25#	6-110	7-69	8-142										
STUFF	8-103	8-116	8-125#	8-138										
TESTCS	1-32#	5-69	5-77											
UNDEF	1-45#													
VALERR	1-66#													
VALFL	3-30	3-35#												
ZERO	5-36	5-59	5-87	5-91#										

SEI 1-3# 2-27



```
                        .CSECT  MCD20B
                        .CSECT  SYMTAB
                        .TITLE  SYMTAB LE SEARCHING AND BUILDING ROUTINE
                        .GLOBL  COMPR,R0,R1,R2,R3,R4,R5,R6,R7,R8
                        .GLOBL  SCRTCH,INDXTG,ITM1TG,VALTG,LSP,STPTR,D12X,D13X
                        .GLOBL  SHNTCT,SBP,TFLGS1,ERRCD,FUDGH,FUDGL,KBFLAG
                        .GLOBL  PRMCOD,NUMCOD,STGCOD,CONCOD,DISBLE,ENABLE,RTRNTG
0080                ATTR1= H80
00C0                ATTR2= H0C0

                    ;       SYMTAB SEARCHES THE SYMBOL (NAME) TABLE AND CREATES NEW
                    ;       ENTRIES IF NECESSARY.
                    ;
                    ;       INPUT:  R3      -NAME TO BE SEARCHED FOR
                    ;               TFLGS1  -STATUS FLAGS.
                    ;
                    ;       OUTPUT: R3      -POINTER TO THE ENTRY FOUND OR CREATED
                    ;
                    ;
                    ;                                       ;NAME IS IN R3, RESULT WILL BE PUT IN R3
0000  CE  000G          LDX     STPTR,I         ;LEX IS USING R2,R6
0003  DF  00G           STX     R0,D            ;R0 WILL BE PTR0
0005  DE  00G           LDX     STPTR,D         ;X WILL BE PTR1
0007  D6  00G           LDA B   TFLGS1,D
0009  C5  20            BIT B    H20,I  ; IS DEFFLG SET?
000B  27  10            BEQ     START           ;IF NOT, SCAN TABLE
000D  96  00G           LDA A   R3,D            ;STORE THE NAME OF THE PARAMETER
000F  A7  02            STA A   2,X             ;IN THE FIRST NT ENTRY
0011  96  01G           LDA A   R3+1,D
0013  A7  03            STA A   3,X
0015  86  08            LDA A    H08,I  ;MARK THIS ENTRY AS A VALID
0017  A7  04            STA A   4,X             ;PARAMETER
0019  1B                ABA
001A  97  00G           STA A   TFLGS1,D                ;AND SET THE PARM BIT
001C  39                RTS
001D  96  00G       START:  LDA A   R3,D
001F  F6  0001G             LDA B   (R3+1)
0022  8C  0000      SCAN:   CPX     0,I             ;PTR1=NULL?  IF SO, CREATE
0025  27  28            BEQ     BUILD           ;A NEW ENTRY
0027  A1  02            CMP A   2,X             ;COMPARE 1ST LETTER OF NAME TO THE NAME
                                                ;IN THIS ENTRY
0029  25  16            BCS     KLUDGE          ;IF NAME) NAME IN ENTRY THEN NO NAME FOR
                                                ;ENTRY EXISTS, NEED TO CREATE ENTRY
002B  22  1C            BHI     SKIP            ;IF NAME(NAME IN ENTRY THEN LOOK AT NEXT
                                                ; ENTRY
002D  E1  03            CMP B   3,X             ;IF FIRST LETTERS ARE THE SAME LOOK AT
                                                ;SECOND
002F  25  10            BCS     KLUDGE          ;LETTER AND DO SAME TESTS
0031  22  16            BHI     SKIP
                                                ;NAMES MATCH, WE'VE FOUND IT
0033  A6  04            LDA A   4,X             ;LOAD THE ATTRIBUTE  BYTE
0035  81  08            CMP A    H08,I          ;IS THIS A PARM ENTRY?
0037  26  05            BNE     NORMEX          ;IF NOT TAKE NORMAL EXIT
0039  9A  00G           ORA A   TFLGS1,D        ;IF SO
003B  97  00G           STA A   TFLGS1,D        ;TURN ON PARM BIT
003D  39                RTS
003E  DF  00G       NORMEX: STX     R3,D            ;STORE PTR1 IN R3
```

```
0040  39                  RTS               ;AND RETURN
0041  A6     04   KLUDGE: LDA A  4,X
0043  85     08           BIT A   H08,I
0045  27     08           BEQ    BUILD
0047  96     00G          LDA A  R3,D
0049  DF     00G  SKIP:   STX    R0,D       ;SET PTR0=PTR1
004B  EE     00           LDX    0,X        ;AND THEN ADVACE PTR1
004D  20     D3           BRA    SCAN    ;AND GO BACK TO EXAMINE THE NEXT ENTRY
004F  86     00G  BUILD:  LDA A  RTRNTG,I
0051  36                  PSH A
0052  BD   0000G          JSR    DISBLE     ;SET NOT REALLY
0055  9F     00G          STS    R7,D       ;R7 STORES SP
0057  D6     01G          LDA B  (LSP+1),D  ;CALC LSP +SLOP+13-SP
0059  96     00G          LDA A  LSP,D
005B  CB     9D           ADD B   H9D,I
005D  89     01           ADC A   H01,I
005F  D0     01G          SUB B  (R7+1),D
0061  92     00G          SBC A  R7,D
0063  2D     17           BLT    ENUF       ;IF HI BYTE < 0 THEN THERE IS ENOUGH
                                            ;ROOM FOR THE NEW ENTRY
0065  2E     04           BGT    FULL       ;IF >0 THEN NOT AND NEED TO DO A
                                            ;MEMORY COMPRESS
0067  C1     00           CMP B  0,I        ;IF =0 THEN TEST LOW BYTE, IF IT IS ZERO
0069  27     11           BEQ    ENUF       ;THERE IS ENOUGH ROOM, ELSE THERE IS NOT.
006B  37          FULL:   PSH B             ;A & B CONTAIN THE MIN # OF BYTES
006C  36                  PSH A             ;TO BE RECOVERED FOR A SUCCESSFUL
006D  86     00G          LDA A  ITM1TG,I   ;MEMORY COMPRESS, PUT THIS NUMBER ON
006F  36                  PSH A             ;THE STACK AND TAG IT.
0070  BD   0000G          JSR    COMPR      ;CALL MEMORY COMPRESS AND IF
0073  96     00G          LDA A  ERRCD,D    ;IT FALLS TO RECOVER ENOUGH MEMORY
0075  27     05           BEQ    ENUF
0077  BD   0000G  EXIT:   JSR    ENABLE
007A  32                  PUL A
007B  39                  RTS
007C  DF     00G  ENUF:   STX    R8,D       ; STORE PTR1 IN R8
007E  DE     00G          LDX    R7,D       ; SUBTRACT 13 FROM THE STACK POINTER
0080  BD   0000G          JSR    D12X
0083  35                  TXS
0084  DE     00G          LDX    SBP,D      ; SUBTRACT 13 FROM STACK BOTTOM
0086  BD   0000G          JSR    D13X       ; POINTER ALSO
0089  DF     00G          STX    SBP,D
008B  30                  TSX
008C  09                  DEX
008D  9C     00G  COPY:   CPX    SBP,D      ;COPY THE STACK DOWN 13. BYTES
008F  27     07           BEQ    FINALY     ;MAKING ROOM FOR THE NEW
                                            ;NT ENTRY
0091  08                  INX
0092  A6     0D           LDA A  13.,X      ;SBP+1 POINTS TO THE FIRST BYTE
0094  A7     00           STA A  0,X        ;OF THE NEW ENTRY
0096  20     F5           BRA    COPY
0098  DE     00G  FINALY: LDX    R0,D       ;SET PTR0->NEXT ENTRY PTR
009A  96     01G          LDA A  (SBP+1),D  ;EQUAL TO NEW ENTRY PTR
009C  8B     01           ADD A  1,I
009E                      SEI
    009E  01   0F   .BYTE  01,17
00A0  A7     01           STA A  1,X
00A2  96     00G          LDA A  SBP,D
00A4  89     00           ADC A  0,I
```

```
129  00A6  A7  00          STA A  0,X
130  00A8  EE  00          LDX    0,X          ;SET NEW ENTRY, NEXT ENTRY
131  00AA  96  01G         LDA A  (R8+1),D     ;POINTER EQUAL TO PTR1
132  00AC  A7  01          STA A  1,X
133  00AE  96  00G         LDA A  R8,D
134  2080  A7  00          STA A  0,X
135  00B2  96  01G         LDA A  (R3+1),D           ;STORE NAME IN NEW
136  00B4  A7  03          STA A  3,X          ;ENTRY
137  00B6  96  00G         LDA A  R3,D
138  00B8  A7  02          STA A  2,X
139  00BA  0E              CLI
140  00BB  86  10          LDA A  #10,I        ;CONSTRUCT ATTRIBUTE BYTE
141  00BD  94  00G         AND A  TFLGS1,D           ;AND STORE IT
142  00BF  26  02          BNE    1$
143  00C1  86  40          LDA A  #40,I
144  00C3  8A  80      1$: ORA A  #80,I
145  00C5  A7  04          STA A  4,X
146  00C7  DF  00G         STX    R3,D
147  00C9  20  AC          BRA    EXIT
148       0001'            .END
```

```
SYMTAB LE SEARCHING AND BUILDIN       RT-11 MMAC  VM02-10   14-OCT-76 01:42:01 PAGE 1+
SYMBOL TABLE

ATTR1 =  0080        ATTR2 =  00C0        BUILD    004FR   03  COMPR = xxxxxx G      CONCOD= xxxxxx G
COPY     0C30R   03  DISBLE= xxxxxx G      D12X  = xxxxxx G      D13X  = xxxxxx G      ENABLE= xxxxxx G
ENUF     007CR   03  ERRCD = xxxxxx G      EXIT     0077R   03  FINALY   0098R   03  FUDGH = xxxxxx G
FUDGL = xxxxxx G     FULL     0068R   03  IMXTG = xxxxxx G      ITM1TG= xxxxxx G      KBFLAG= xxxxxx G
KLUDGE   0041R   03  LSP  = xxxxxx G      NORMEX   003ER   03  NUMCOD= xxxxxx G      PRMCOD= xxxxxx G
RTRNTG= xxxxxx G     R0   = xxxxxx G      R1   = xxxxxx G      R2   = xxxxxx G      R3   = xxxxxx G
R4   = xxxxxx G      R5   = xxxxxx G      R6   = xxxxxx G      R7   = xxxxxx G      R8   = xxxxxx G
SBP  = xxxxxx G      SCAN     0022R   03  SCRTCH= xxxxxx G     SWITCT= xxxxxx G      SKIP     0049R   03
START    001DR   03  STGCOD= xxxxxx G      STPTR = xxxxxx G      TFLGS1= xxxxxx G      VALTG = xxxxxx G

  ABS.   0000    00
         0000    01
MCD20B   0000    02
SYMTAB   00CB    03
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  3272.  WORDS
.SY:SYMTAB/C<DK1:SEICLI.SYMTAB
```

CROSS REFERENCE TABLE (CREF V01-03.)

```
ATTR1   1-23#
ATTR2   1-24#
BUILD   1-54    1-76    1-81#
COMPR   1-19#   1-101
CONCOD  1-22#
COPY    1-116#  1-121
D12X    1-20#   1-109
D13X    1-20#   1-112
DISBLE  1-22#   1-83
ENABLE  1-22#   1-104
ENUF    1-91    1-96    1-103   1-107#
ERRCD   1-21#   1-102
EXIT    1-104#  1-147
FINALY  1-117   1-122#
FUDGH   1-21#
FUDGL   1-21#
FULL    1-93    1-97#
IMXTG   1-20#
ITM1TG  1-20#   1-99
KBFLAG  1-21#
KLUDGE  1-57    1-63    1-74#
LSP     1-20#   1-85    1-86
NORMEX  1-68    1-72#
NUMCOD  1-22#
PRMCOD  1-22#
R0      1-19#   1-37*   1-78*   1-122
R1      1-19#
R2      1-19#
R3      1-19#   1-42    1-44    1-51    1-52    1-72*   1-77    1-135   1-137   1-146*
R4      1-19#
R5      1-19#
R6      1-19#
R7      1-19#   1-84*   1-89    1-90    1-108
R8      1-19#   1-107*  1-131   1-133
RTRNTG  1-22#   1-81
SBP     1-21#   1-111   1-113*  1-116   1-123   1-127
SCAN    1-53#   1-80
SCRTCH  1-20#
SHNTCT  1-21#
SKIP    1-5?    1-64    1-78#
START   1-41    1-51#
STGCOD  1-22#
STPTR   1-20#   1-36    1-38
TFLGS1  1-21#   1-39    1-49*   1-69    1-70*   1-141
VALTG   1-20#
```

SEI 1-3# 1-125

```
                                .TITLE  TYPIN   KEYBOARD INPUT EDITOR
                                .IDENT  /BE8045/
                        ;
                        ;       THIS IS THE KEYBOARD INPUT EDITOR.  THE FUNCTIONS ARE
                        ;       DEFINED IN THE HEADERS FOR EACH SECTION OF CODE.
                        ;
0000                            .ASECT                  ;GOES BELOW PIAS FOR NOW
8000                              = H8000
                        ;
                                .GLOBL  TYPIN
                        ;
0080            BLOT    =       H080            ;SCREEN CHAR TO BLOT OUT OLD CHARACTER
                                .GLOBL  KBFLAG
0040            ENDKEY  =       H40             ;FLAG FOR LINE READY
0020            BFRSTT  =       H20             ; STATUS = VALID BUFFER
0008            ATLOD   =       H08             ; AUTO LOAD KEY
0001            INPUTK  =       H01             ; INPUT ACTIVE FLAG
                                .GLOBL  CRSTAT          ; CR STATUS FOR BUFFER MANAGER
0001            CRNORM  =       1               ;
0003            PGMFP   =       3               ;FOWARD POINTER IN PROGRAM LIST
0007            PGMLNN  =       7               ;LINE NUMBER FIELD
                                .GLOBL  A.PRIM          ; I/O CONTROL BYTE
                                .GLOBL  A.STAT          ; STATUS OF CHAN A
                                .GLOBL  MTSREG          ; MAG STATUS REGISTER
                                .GLOBL  A.PTR
                                .GLOBL  A.END           ;LAST TEXT LOCATION+1
                                .GLOBL  A.MAX           ;LAST LOCATION IN BUFFER
                                .GLOBL  FPC             ;WORK AREA IN F.P. PACK
                                .GLOBL  ERRCD
                                .GLOBL  PGMPTR,CLPTR,NLPTR
                                .GLOBL  XAXIS
                                .GLOBL  EDTEND          ;ADDRESS OF LAST TEXT LOCATION+1
                                .GLOBL  EDTBFR          ;EDIT BUFFER
                                .GLOBL  EDTPTR          ;CURRENT BUFFER POINTER (CURSOR)
                                .GLOBL  EDTMAX          ;LAST NONE BLANK IN LINE+1 AFTER BCKSCN
                                .GLOBL  ANCRNT          ;AUTO NO. CURRENT VALUE
                                .GLOBL  ANINCR          ;AUTO NO. INCR (IF = 0 AUTO NO. IS RESET)
                                .GLOBL  ANHOLD          ;DON'T AUTO NUMBER THIS TIME
                                .GLOBL  UNCOMP
                                .GLOBL  INAGTE          ;INTERGER TO ASCI
                                .GLOBL  AFPITI          ;ASCI TO FPN
                                .GLOBL  FIX1
                                .GLOBL  DSPCHR,CTLCHR,TMPHX
                                .GLOBL  R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R15,R16,ZX
                                .GLOBL  SCRTCH          ;SYSTEM WORK SPACE
                                .GLOBL  MTAFIN          ;MAG TAPE FIND ROUTINE
                                .GLOBL  INITMT          ;TO INIT MAG TAPE
                                .GLOBL  BFRALC          ;BUFFER ALLOCATION ROUTINE
                                .GLOBL  ADRDEV          ;ADDRESS DEVICE
                                .GLOBL  UNADR           ;INVERSE
                                .GLOBL  FIND,OPRADR     ;FAKE OUT STUFF
                                .GLOBL  CTKN,OLDCOD     ;SET UP CONYROLS FOR OLD
                                .GLOBL  GLBFLG          ;FOR SETTING RUN MODE
0080            RUNFLG  =       H80
                                .GLOBL  OLDONE          ;FOR AUTO LOAD TO DO OLD
                                .GLOBL  MTSTAT          ;MAG TAPE STATUS BYTE
                                .GLOBL  MTSASC          ;  ASCII FILE
```

```
   73                                   .GLOBL  MTSPGM          ; PROGRAM FILE
```

```
                                        .MACRO TBL     CHR,CODE,RTN
                                CHR     =       ^O'CODE
                                        .BYTE   CHR
                                        .WORD   RTN
                                        .ENDM
                                ;
                                ;
                                ;   ENTRY IS A SPECIAL CHARACTER FOLLOWED BY POINTER TO ROUTINE.
                                ;
8000                            TITBL:  TBL     NMBS,88,TBSTP
8003                                    TBL     CR,0D,TCR
8006                                    TBL     NMCR,8D,TCR
8009                                    TBL     RUBOUT,7F,TBDLT
800C                                    TBL     SPACE,20,TFSTP
800F                                    TBL     EXPLN,F0,TEXPLN
8012                                    TBL     CMPLN,B0,TCMPLN
8015                                    TBL     BSTP,F1,TBSTP
8018                                    TBL     BDLT,B1,TBDLT
8018                                    TBL     FSTP,F2,TFSTP
801E                                    TBL     FDLT,B2,TFDLT
8021                                    TBL     CLRLN,F3,TCLRLN
8024                                    TBL     RLN,B3,TRCLN
8027                                    TBL     PRC,F4,TPRC
802A                                    TBL     NPRC,B4,TNPRC
802D                                    TBL     AN,F5,TAN
8030                                    TBL     ATLD,F7,TATLD
8033                            TITBLX  =       .               ;END OF TABLE
                                ;
0008                            BS      =       ^O08
```

```
                                ;
                                ;
                                ;       MAIN ENTRY POINT FOR LINE EDITOR
                                ;
                                ;       INPUT CHARACTER IS IN ACC-A
                                ;
8033    CE  8000        TYPIN:  LDX     TITBL,I         ;SEE IF THIS IS A CONTROL FUNCTION.
8036    C6  11                  LDA B   TITBLX-TITBL/3,I
                        
                        
8038    A1  00          INLPA:  CMP A   0,X             ;INPUT CHARACTER IN ACC-A.
803A    27  2E                  BEQ     EDTKEY          ;IF FOUND, BRANCH.
803C    08                      INX                     ;ENTRIES ARE
803D    08                      INX                     ;   THREE
803E    08                      INX                     ;   BYTES LONG.
803F    5A                      DEC B
8040    26  F6                  BNE     INLPA           ;IF NOT IN TABLE, IT IS TEXT.
                                ;
8042    DE  00G         NEWCHR: LDX     EDTPTR,D        ;PUT A NEW CHAR INTO BUFFER
8044    26  0E                  BNE     INSKPB          ;IF EDTPTR=0 BFR IS EMPTY
8046    DE  00G                 LDX     EDTEND,D        ;CLEAR IT IF NEW LINE IS STARTING
8048    C6  20                  LDA B   SPACE,I
804A    E7  00          INLPB:  STA B   0,X
804C    09                      DEX
804D    8C  000u                CPX     EDTBFR,I
8050    26  F8                  BNE     INLPB
8052    DF  00G                 STX     EDTPTR,D        ;SET UP NEW EDTPTR
8054    84  7F          INSKPB: AND A   127.,I          ;FORGET HIGH ORDER BIT
8056    A7  00                  STA A   0,X             ;PUT NEW CHARACTER INTO BUFFER
8058    BD  0000G               JSR     DSPCHR          ;PRINT CHAR ON SCREEN
805B    DE  00G                 LDX     EDTPTR,D        ;UPDATE PTR (CURSOR LOCATION)
805D    08                      INX
805E    9C  00G                 CPX     EDTEND,D        ;WILL A NEW CHAR FIT
8060    27  03                  BEQ     INSKPC          ;IF NOT TRA
8062    DF  00G                 STX     EDTPTR,D        ;NEW CURSOR LOC
8064    39              RTSA:   RTS
                                ;
8065    86  08          INSKPC: LDA A   BS,I            ;BACK UP SCREEN CURSOR
8067    7E  0000G               JMP     CTLCHR          ;SCREEN CONTROL & RTSA
                                ;
806A    86  20          EDTKEY: LDA A   SPACE,I         ;A LOT OF ROUTINES CAN USE BLANK
806C    EE  01                  LDX     1,X             ;I FOUND A SPECIAL FUNCTION CHARACTER!!!
806E    6E  00                  JMP     0,X             ;GO AND GET CODE FOR  IT.
```

```
                                        ;
                                        ;       CARRIGGE RETURN FUNCTION
                                        ;       SETS UP END OF BUFFER AND I/O CONTROLS.
                                        ;       ALSO RESETS EDITOR CONTROLS TO START NEW LINE NEXT ENTRY
                                        ;
8070    DE      000     TCR:    LDX     EDTPTR,D        ;IS LINE BLANK
8072    27      35              BEQ     CRNULL          ;SET UP NULL LINE
8074    08                      INX                     ;CURSOR NEVER GETS TO EDTEND
8075    9C      000             CPX     EDTEND,D        ;IS CURSOR AT END OF BUFFER
8077    27      0C              BEQ     CRSKIP          ;IF SO DON'T MOVE IT
8079    09                      DEX                     ;I NEED TO LOOK AT CURSOR NOW
807A    A1      00      CRLOOP: CMP A   0,X             ;IS THERE A BLANK
807C    26      08              BNE     CRMOVE          ;IF IN DATA MOVE CURSOR
807E    08                      INX                     ;TRY NEXT ONE
807F    9C      000             CPX     EDTEND,D        ;END OF DATA YET
8081    26      F7              BNE     CRLOOP
8083    DE      000             LDX     EDTPTR,D        ;PUT CR AT CURSOR
                                        ;
8085    DF      000     CRSKIP: STX     EDTMAX,D        ;RESET END PTR
8087    20      02              BRA     CRCONT          ;JOIN COMMON CODE
                                        ;
8089    8D      75      CRMOVE: BSR     BCKSCN          ;FIND END OF BUFFER
808B    86      0D      CRCONT: LDA A   CR,I
808D    A7      00              STA A   0,X
808F    DF      000             STX     A.END,D         ;SET UP I/O POINTER TO END
8091    CE      0000G           LDX     EDTBER,I        ;SET UP I/O
8094    DF      000             STX     A.PTR,D         ;    POINTERS.
8096    D6      000             LDA B   A.STAT,D        ;SET BUFFER VALID
8098    CA      20              ORA B   BFRSTT,I
809A    D7      000             STA B   A.STAT,D
809C    D6      000             LDA B   KBFLAG,D        ;INPUT READY
809E    CA      40              ORA B   ENDKEY,I        ;    FLAG
80A0    D7      000             STA B   KBFLAG,D        ;    FOR IOPROC.
80A2    C6      01              LDA B   CRNORM,I        ; MARK BUFFER FOR GETCHR
80A4    D7      000             STA B   CRSTAT,D
80A6    7E      8132            JMP     TCLRLN          ;RESET CONTROLS & SEND CR TO SCREEN
                                        ;
80A9    CE      0000G   CRNULL: LDX     EDTBER,I        ;LINE HAS ONLY CR & ONE BYTE ZERO
80AC    DF      000             STX     EDTMAX,D        ;SAVE SO RECALL WILL WORK
80AE    20      DB              BRA     CRCONT
```

```
                                ;
                                ;       DELETE CHARACTER AND STEP FORWARD
                                ;
808O    DE    00G       TFOLT:  LDX     EDTPTR,O        ;GET CURSOR LOCATION
8082    27    8O        TFOLT:  BEQ     RTSA            ;IF LINE IS EMPTY EXIT NOW
8084    A1    00              CMP A   0,X             ;AM I AT A BLANK
8086    26    28              BNE     DELETE          ;BLOT NOW
8088    8D    02              BSR     TFSTP           ;GO SPACE FOWARD
808A    20    24              BRA     DELETE          ;NOW BLOT
                                ;
                                ;       STEP FOWARD - SAME AS SPACE KEY
                                ;
808C    DE    00G       TFSTP:  LDX     EDTPTR,O        ;GET CURSOR LOC
808E    27    02              BEQ     FSTPSK          ;IF THIS IS NEW LINE USE BLANK
8090    A6    00              LDA A   0,X             ;ECHO OLD DATA
8092    7E    8042    FSTPSK: JMP     NEWCHR
                                ;
                                ;       DELETE CHARACTER AND BACK STEP - SAME AS RUBOUT KEY
                                ;
80C5    DE    00G       TBOLT:  LDX     EDTPTR,O        ;CURSOR LOC
80C7    27    98              BEQ     RTSA            ;NO LINE - NO OP
80C9    A1    00              CMP A   0,X             ;IS CURRENT LOC A BLANK
80CB    26    13              BNE     DELETE
80CD    8D    22              BSR     BACKUP          ;BACK SPACE
80CF    20    0F              BRA     DELETE
                                ;
                                ;       BACK STEP FUNCTION - SAME AS BACK SPACE KEY
                                ;
8001    DE    00G       TBSTP:  LDX     EDTPTR,O        ;GET CURSOR LOC
8003    27    8F              BEQ     RTSA            ;NOTHING TO NEW
8005    8D    1A              BSR     BACKUP          ;BACK SPACE
8007    DE    00G             LDX     EDTPTR,O        ;GET CHAR TO ECHO
8009    A6    00              LDA A   0,X
8008    BD    0000G           JSR     DSPCHR
800E    20    1B              BRA     BCKUPA          ;I MUST HAVE SPACE TO BACK UP OVER
```

```
                                ;
                                ;       INTERNAL BLOT ROUTINE
                                ;
80E0  0E  000   DELETE: LDX     EDTPTR,D        ;AM I LOOKING AT A BLANK
80E2  86  20            LDA A   SPACE,I         ;ACC-A MAY HAVE BEEN DESTROYED
80E4  A1  00            CMP A   0,X
80E6  27  3C            BEQ     RTSB
80E8  A7  00            STA A   0,X             ;REPLACE CHAR IN BFR
80EA  86  80            LDA A   BLOT,I          ;PUT BLOT ON SCREEN
80EC  BD  0000G         JSR     DSPCHR
80EF  20  0A            BRA     BCKUPA          ;BACK UP SCREEN CURSOR
                                ;
                                ;       BACKUP CURSOR ONE LOCATION
                                ;
80F1  0E  000   BACKUP: LDX     EDTPTR,D        ;SEE IF I CAN DO IT
80F3  8C  0000G         CPX     EDTBFR,I
80F6  27  2C            BEQ     RTSB            ;NO WAY
80F8  09                DEX                     ;BACK UP TYPINS CURSOR
80F9  0F  000           STX     EDTPTR,D
80FB  86  08    BCKUPA: LDA A   BS,I            ;BACK UP SCREEN
80FD  7E  0000G         JMP     CTLCHR
                                ;
                                ;       BACK SCAN FUNCTION
                                ;
8100  C6  20    BCKSCN: LDA B   SPACE,I         ;GET BLANK FOR SCAN
8102  0E  000           LDX     EDTEND,D        ;SCAN FROM RIGHT TO LEFT
8104  09       BSCNLP: DEX
8105  E1  00            CMP B   0,X             ;DID I FIND NON BLANK
8107  26  00            BNE     BSCNB
8109  9C  000           CPX     EDTPTR,D        ;IF I PASS EDTPTR (CURSOR) STOMP ON IT
810B  26  03            BNE     BSCNSK
810D  7F  0000G         CLR     EDTPTR          ;THIS WILL CAUSE RESET LATER
8110  8C  0000G BSCNSK: CPX     EDTBFR,I        ;FALL OFF FRONT?
8113  26  EF            BNE     BSCNLP          ;I HAVE LAST ONE
8115  09                DEX                     ;EASY WAY TO FAKE OUT INX I DON'T NEED
                                ;
8116  08       BSCNA:  INX                     ;SET UP & SAVE END ADDR
8117  0F  000           STX     EDTMAX,D
8119  96  000           LDA A   EDTPTR,D        ;IS CURSOR IN NEED OF AN ADDRESS
811B  26  07            BNE     RTSB            ;NO
811D  9C  000           CPX     EDTEND,D        ;DON'T LET PTR = END
811F  26  01            BNE     BSCNB
8121  09                DEX
8122  0F  000   BSCNB:  STX     EDTPTR,D
8124  39       RTSB:   RTS
```

```
                                        ;
                                        ;       AUTO LOAD KEY
                                        ;
8125  96  000   TATLD:  LDA A  KBFLAG,D      ;SET UP SYSTEM CONTROL BITS
8127  85  01            BIT A  INPUTK,I
8129  26  F9            BNE    RTSB
812B  8A  08            ORA A  A7.00,I      ;THE SYSTEM WILL DO AUTO LOAD LATER
812D  97  000           STA A  KBFLAG,D     ;SINCE IT IS NOT SAFE NOW
812F  BD  822F          JSR    EDTCLS       ;CLOSE BUFFER AND DO CR
                        ;
                        ;       CLEAR LINE FUNCTION - ALSO SAVE SO RECALL WILL WORK IF DATA
                        ;       IS STILL IN BUFFER (NO NEW DATA HAS BEEN SEEN).
                        ;
8132  DE  000   TCLRLN: LDX    ZX,D         ;RESET CURSOR FOR EMPTY BFR FLAG
8134  DF  000           STX    EDTPTR,D
8136  86  00    DOCR:   LDA A  CR,I         ;DO CR ALSO
8138  7E  0000G         JMP    CTLCHR
                        ;
                        ;       RECALL LINE BUFFER
                        ;
813B  DE  000   TRCLN:  LDX    EDTPTR,D
813D  26  00            BNE    PRINT
813F  DE  000           LDX    EDTMAX,D     ;IF = 0 BFR IS EMPTY (POWER UP STATE)
8141  27  EF            BEQ    TCLRLN
8143  8C  0000G         CPX    EDTBFR,I     ;IF BUFFER IS EMPTY SKIP OUT
8146  27  EA            BEQ    TCLRLN       ;IT LOOKS LIKE NULL RECALL
8148  A7  00            STA A  0,X
814A  A7  01            STA A  1,X
                        ;
                        ;       PRINT - THIS IS INTERNAL FUNCTION TO PRINT EDIT BUFFER.
                        ;       START AT EDTBFR PRINT TO EDTMAX AND MOVE CURSOR TO EDTPTR.
                        ;
814C  BD  8136  PRINT:  JSR    DOCR         ;SEND CR TO SCREEN
814F  BD  AF    PRINTN: BSR    BCKSCN       ;CLEAN UP BFR
8151  CE  0000G         LDX    EDTBFR,I     ;START AT BFR ORIGIN
8154  DF  000   PRINTL: STX    R0,D
8156  9C  000           CPX    EDTMAX,D
8158  27  0A            BEQ    PRINTJ
815A  A6  00            LDA A  0,X          ;GET CURRENT CHARACTER.
815C  BD  0000G         JSR    DSPCHR       ;OUTPUT CHARACTER
815F  DE  000           LDX    R0,D
8161  08               INX
8162  20  F0            BRA    PRINTL
8164  DF  000   PRINTJ: STX    R0,D
8166  9C  000           CPX    EDTPTR,D     ;SEE IF I AM AT LOGICAL CURSOR
8168  27  BA            BEQ    RTSB         ;IF LOCATION IS CORRECT STOP
816A  86  08            LDA A  BS,I
816C  BD  0000G         JSR    CTLCHR       ;OUTPUT A BACKSPACE TO SCREEN.
816F  DE  000           LDX    R0,D
8171  09               DEX                 ;DECREMENT POINTER
8172  20  F0            BRA    PRINTJ
```

```
                                        ;
                                        ;       EXPAND LINE FUNCTION - MOVE ALL CHARACTERS TO RIGHT OF CURSOR
                                        ;       TO FAR RIGHT OF BUFFER AND FILL WITH SPACES
                                        ;
8174  8D  29  TEXPLN: BSR   ANYCHR       ;TEST FOR ALL BLANKS TO RIGHT OF CURSOR
8176  DE  00G EXPLNL: LDX   EDTEND,D     ;SEE IF LAST CHAR IS BLANK
8178  09              DEX
8179  A1  00          CMP A  0,X
817B  26  CF          BNE   PRINT       ;IF NOT PRINT BUFFER
817D  09        EXPLNM: DEX
817E  E6  00          LDA B  0,X        ;MOVE CHAR IN BFR RIGHT ONE
8180  E7  01          STA B  1,X
8182  9C  00G         CPX   EDTPTR,D    ;AM I DONE
8184  26  F7          BNE   EXPLNM
8186  A7  00          STA A  0,X        ;PUT SPACE AT CURSOR
8188  20  EC          BRA   EXPLNL      ;GO TO MAIN LOOP
                                        ;
                                        ;       COMPRESS LINE FUNCTION - DELETE ALL SPACE FROM CURSOR TO NONE
                                        ;       BLANK AND PRINT LINE.
                                        ;
818A  8D  13  TCMPLN: BSR   ANYCHR      ;TEST FOR NO DATA TO RIGHT OF CURSOR
818C  DE  00G CMPLNL: LDX   EDTPTR,D    ;GET CURSOR LOC
818E  A1  00          CMP A  0,X        ;IS THERE SPACE @ CURSOR
8190  26  BA          BNE   PRINT       ;TIME TO PRINT IT
8192  E6  01  CMPLNM: LDA B  1,X        ;MOVE ONE LOCATION LEFT
8194  E7  00          STA B  0,X
8196  08              INX
8197  9C  00G         CPX   EDTEND,D    ;AT END OF BUFFER
8199  26  F7          BNE   CMPLNM
819B  A7  00          STA A  0,X        ;SET LAST LOCATION TO BLANK
819D  20  ED          BRA   CMPLNL      ;CALL IT AGAIN
```

```
                                ;
                                ;       SEE IF BUFFER IS BLANKS TO RIGHT OF CURSOR
                                ;
819F  DE  00G   ANYCHR: LDX     EDTPTR,D        ;GET CURSOR
81A1  27  0D            BEQ     RTSTWO          ;HALT PROCESS IF NO DATA IN BFR
81A3  A1  00    ANYCLP: CMP A   0,X             ;LOOK FOR NONE BLANK
81A5  26  0B            BNE     RTSB2
81A7  08                INX
81A8  9C  00G           CPX     EDTEND,D        ;AT END OF BUFFER YET
81AA  26  F7            BNE     ANYCLP
81AC  32                PUL A
81AD  32                PUL A
81AE  20  9C            BRA     PRINT           ;OUTPUT LINE ONLY
                        ;
8180  32        RTSTWO: PUL A                   ;DUMP ONE RETURN ADDR
8181  32                PUL A
8182  39        RTSB2:  RTS                     ;GO TO ORIGINAL CALLER
```

```
                                ;
                                ;       NEXT PROGRAM LINE RECALL
                                ;
81B3  86  00      TNPRC:  LDA A   CR,I             ;SEE IF MORE THAN ONE CR
81B5  B1  0000G           CMP A   EDTBER
81B8  27  30              BEQ     ERROR           ;IF SO FORGET IT
81BA  8D  5A              BSR     PNMBR           ;PRIME CONTROLS & GET NUMBER
81BC  96  00G             LDA A   ERRCD,D         ;DID I GET A NUMBER
81BE  26  37              BNE     ERROR           ;NO
81C0  08                  INX                     ;GET NEXT NUMBER
81C1  26  0A              BNE     PRCOK           ;IF OKAY GO ON
                                ;
                                ;       PROGRAM LINE RECALL
                                ;
81C3  DE  00G     TPRC:   LDX     EDTPTR,D
81C5  27  67              BEQ     RTSC
81C7  8D  40              BSR     PNMBR           ;PRIME CONTROLS & GET NUMBER
81C9  96  00G             LDA A   ERRCD,D         ;DID IT WORK
81CB  26  2A              BNE     ERROR           ;NO
81CD  DF  00G     PRCOK:  STX     R0,D            ;MOVE X TO R0
81CF  96  00G             LDA A   XAXIS,D         ;CAN HE READ IT
81D1  26  24              BNE     ERROR           ;NICE CLEAN EXIT STAGE RIGHT
81D3  96  00G             LDA A   R0,D
81D5  D6  01G             LDA B   R0+1,D
81D7  DE  00G             LDX     PGMPTR,D        ;LOOK FOR LINE IN PGM LIST
81D9  27  53      PRCLPA: BEQ     RTSC            ;IF END OF LIST FORGET IT
81D8  A1  07              CMP A   PGMLNM,X        ;IF N > LINE NO. LOOP
81DD  22  06              BHI     PRCSKA
81DE  25  08              BCS     PRCFND
81E1  E1  08              CMP B   PGMLNM+1,X
81E3  23  04              BLS     PRCFND
81E5  EE  03      PRCSKA: LDX     PGMEP,X         ;RUN DOWN CHAIN
81E7  20  F0              BRA     PRCLPA
                                ;
81E9  DF  00G     PRCFND: STX     R1,D            ;SET UP FOR UNCOMPILE
81EB  BD  0000G           JSR     UNCOMP
81EE  DE  00G             LDX     ZX,D            ;RESET CURSOR LOCATION
81F0  DF  00G             STX     EDTPTR,D
81F2  8D  07              BSR     EDTCLR          ;CLEAN UP REST OF BUFFER
81F4  7E  814C            JMP     PRINT           ;PRINT IT NOW
                                ;
81F7  7F  0000G   ERROR:  CLR     ERRCD
81FA  39                  RTS
```

```
                                ;
                                ;       ROUTINE TO CLEAR BUFFER AFTER UNCOMP
                                ;
                                        .GLOBL  EDTCLR
81FB  DE  006   EDTCLR: LDX     A.END,D         ;ADDR OF LAST CHAR
81FD  09                DEX
81FE  9C  006           CPX     EDTEND,D
8200  27  2C            BEQ     RTSC
8202  08                INX
8203  86  20            LDA A   SPACE,I
8205  C6  0D            LDA B   CR,I            ;MAY BE CR AT A.END
8207  E1  00            CMP B   0,X
8209  26  02            BNE     CLRLP
8208  A7  00            STA A   0,X             ;CHANGE CR TO SPACE
8200  A7  01    CLRLP:  STA A   1,X
820F  9C  006           CPX     EDTEND,D        ;ALL DONE
8211  27  18            BEQ     RTSC
8213  08                INX
8214  20  F7            BRA     CLRLP
                                ;
                                ;       ROUTINES TO CONVERT ASCI TO FIXED POINT NUMBERS
                                ;
8216  CE  0000G PNMBR:  LDX     EDTBER,I        ;SET UP CONTROL VARIABLES
8219  DF  006           STX     R0,D
8218  DE  006   GNMBR:  LDX     EDTEND,D
821D  DF  006           STX     R1,D
821F  BD  0000G         JSR     AFPITT          ;CALL ASCI TO FPN
8222  CE  FFFFG         LDX     FPC-1,I         ;PARM FOR FIX1
8225  BD  0000G         JSR     F.IX1
8228  FE  0002G         LDX     FPC+2           ;PASS ANSWER BACK IN X
822B  26  01            BNE     RTSC            ;FORCE NONE ZERO LINE NUMBER
822D  08                INX
822E  39          RTSC: RTS
                                ;
                                ;       EDTCLS - CLOSE THE EDIT BUFFER
                                ;
                                        .GLOBL  EDTCLS
822F  DE  006   EDTCLS: LDX     ZX,D
8231  DF  006           STX     ANINCR,D        ;STOP  AUTO NUMBERING
8233  DF  006           STX     EDTPTR,D        ;AND STOMP ON THE BUFFER
8235  DF  006           STX     EDTMAX,D
8237  39                RTS
```

```
                                ;
                                ;       AUTO NUMBER KEY
                                ;
8238  96  000   TAN:   LDA A   KBFLAG,D      ;IF IN INPUT FOR GET IT
823A  85  01            BIT A   INPUTK,I
823C  26  F0            BNE     RTSC
823E  DE  000           LDX     ANINCR,D    ;IF INCREMENT=0 AUTO NO IS NOT ACTIVE
8240  27  07            BEQ     ANSET
8242  DE  000           LDX     ZX,D        ;RESET CONTROLS
8244  DF  000           STX     ANINCR,D
8246  7E  8132          JMP     TCLRLN      ;CLEAR LINE NOW
                                ;
8249  DE  000   ANSET:  LDX     EDTPTR,D    ;IF LINE IS EMPTY USE DEFAULT
824B  27  06            BEQ     ANFLA
824D  8D  C7            BSR     PNMBR       ;START NUMBER ROUTINE
824F  96  000           LDA A   ERRCD,D     ;DID IT WORK
8251  27  03            BEQ     ANSKPA      ;IF SO USE HIS PARAMETERS
8253  CE  0064  ANFLA:  LDX     100.,I      ;DEFAULT STARTING LINE NUMBER
8256  DF  000   ANSKPA: STX     ANCRNT,D
8258  09              DEX
8259  DF  000           STX     R10,D       ;FOR LATER USE AS OUTPUT NUMBER
825B  DE  000           LDX     EDTPTR,D
825D  27  09            BEQ     ANFLB
825F  96  000           LDA A   ERRCD,D     ;IF ALREADY SET FORGET IT
8261  26  06            BNE     ANFLB
8263  8D  B6            BSR     GNXBR       ;GET NEXT NUMBER
8265  96  000           LDA A   ERRCD,D
8267  27  03            BEQ     ANSKPB
8269  CE  000A  ANFLB:  LDX     10.,I       ;DEFAULT INCREMENT
826C  DF  000   ANSKPB: STX     ANINCR,D
826E  7F  0000G         CLR     ANHOLD
8271  7F  0000G         CLR     ERRCD       ;RESET ERROR CODE BYTE
8274  BD  8132          JSR     TCLRLN      ;NEED CLEAN LINE TO WORK IN
8277  20  1D            BRA     ANCOM       ;GO JOIN COMMON CODE
```

```
                        ;
                        ;       AUTO NUMBER OPERATION - THIS IS CALLED BY MONITOR TO PRINT DIGITS
                        ;
                        .GLOBL  AUTONO
8279    7D  0000G   AUTONO: TST     ANHOLD          ;IS BUFFER OPEN (NETER SYNTAX ERROR)
827C    26  33              BNE     ANSTOP          ;SKIP OUT IF SO
827E    DE  00G             LDX     ANINCR,D        ;IF ZERO AUTO NUMBER IS NOT ACTIVE
8280    27  AC              BEQ     RTSC
8282    DE  00G             LDX     ZX,D            ;PRESET EDITOR CONTROLS FOR LATER CALLS
8284    DF  00G             STX     EDTPTR,D
8286    96  00G             LDA A   ANCRNT,D
8288    D6  01G             LDA B   ANCRNT+1,D
828A    97  00G             STA A   R10,D           ;SET UP OLD LINE NO. SAVE AREA
828C    D7  01G             STA B   R10+1,D
828E    DB  01G             ADD B   ANINCR+1,D
8290    99  00G             ADC A   ANINCR,D
8292    97  00G             STA A   ANCRNT,D        ;SAVE NEW RESULT
8294    D7  01G             STA B   ANCRNT+1,D
8296    96  00G   ANCOM:    LDA A   R10,D           ;GET OLD LINE NO.
8298    D6  01G             LDA B   R10+1,D
829A    7D  0000G           TST     XAXIS
829D    26  48              BNE     ANNEWL          ;IF NOT JUST GIVE NEW NUMBER
829F    DF  00G             LDX     PGMPTR,D        ;SEE IF I HAVE LINES TO PRINT
82A1    27  44      ANLPA:  BEQ     ANNEWL          ;IF END OF LIST THIS IS NEW LINE NO.
82A3    A1  07              CMP A   PGMLNN,X        ;IF OLD >= LINE NO. LOOP
82A5    22  06              BHI     ANSKA
82A7    25  0C              BCS     ANFNDA
82A9    E1  08              CMP B   PGMLNN+1,X
82AB    25  08              BCS     ANFNDA
82AD    EE  03      ANSKA:  LDX     PGMFP,X         ;CHAIN DOWN PROGRAM LIST
82AF    20  F0              BRA     ANLPA
                        ;
82B1    7F  0000G   ANSTOP: CLR     ANHOLD          ;ONLY SKIP AUTO NUMBER ONCE
82B4    39                  RTS                     ;SEE YOU LATER
                        ;
82B5    DF  00G     ANFNDA: STX     R11,D           ;SAVE ADDR OF FOUND LINE
82B7    96  00G             LDA A   ANCRNT,D        ;NOW FIND LINES IN BETWEEN OLD AND NEW
82B9    D6  01G             LDA B   ANCRNT+1,D      ;  LINE NUMBERS
82BB    A1  07              CMP A   PGMLNN,X        ;IF < LINE NO. PRINT LINE - LOOP
82BD    25  28              BCS     ANNEWL          ;IF = LINE NO. PRINT FOR EDIT
82BF    22  06              BHI     ANSKC
82C1    E1  08              CMP B   PGMLNN+1,X
82C3    25  22              BCS     ANNEWL
82C5    27  38              BEQ     ANFNDJ
82C7    DF  00G     ANSKC:  STX     R1,D            ;SET UP FOR UNCOMPILE
82C9    BD  0000G           JSR     UNCOMP
82CC    CE  0000G           LDX     EDTBER,I        ;CLEAN UP LINE
82CF    09                  DEX                     ;FIX FUNNY BUFFER POINTERS
82D0    DF  00G     ANLPC:  STX     R0,D            ;PRINT LINE I GOT
82D2    A6  01              LDA A   1,X
82D4    BD  0000G           JSR     DSPCHR
82D7    DE  00G             LDX     R0,D
82D9    08                  INX
82DA    9C  00G             CPX     A.END,D         ;AM I DONE
82DC    26  F2              BNE     ANLPC
82DE    BD  8136            JSR     DOCR            ;SET UP SCREEN
82E1    DE  00G             LDX     R11,D           ;TRY NEXT LINE
```

```
                                ;
82E3  EE  03          LDX   PGMFP,X
82E5  26  CE          BNE   ANENDA
                                ;
82E7  DE  00G   ANNEWL: LDX   ANCRNT,D     ;SET UP FOR INT TO ASCII
82E9  BD  0000G         JSR   INAGTE       ;CONVERT NUMBER
82EC  CE  0001G         LDX   R16+1,I      ;NOW OUTPUT NUMBER
82EF  DF  00G   ANLPD:  STX   R15,D
82F1  A6  0G            LDA A 0,X
82F3  27  08            BEQ   ANSPAC
82F5  BD  8033          JSR   TYPIN
82F8  DE  00G           LDX   R15,D
82FA  08                INX
82FB  20  F2            BRA   ANLPD
                                ;
82FD  86  20    ANSPAC: LDA A SPACE,I      ;PUT ON ONE SPACE
82FF  7E  8033          JMP   TYPIN        ;IT WILL RETURN TO CALLER
                                ;
8302  DF  00G   ANFNDJ: STX   R1,D         ;ADDR OF LINE TO EDIT
8304  BD  0000G         JSR   UNCOMP
8307  BD  81EB          JSR   EDTCLR       ;CLEAR REST OF BUFFER
830A  7E  814F          JMP   PRINTN
```

```
                                ;
                                ;       AUTO LOAD FUNCTION CALLED FROM IDLE LOOP
                                ;
                                        .GLOBL  ATLOAD
                                ;
8300    7F      0000G   ATLOAD: CLR     A.STAT
                        ;*****************************************************
8310    7F      0000G           CLR     MTSREG          ;*** RESET MAG TAPE STATUS ***
                        ;*****************************************************
8313    CE      0000G           LDX     FIND,I
8316    DF      00G             STX     OPRADR,D
8318    BD      0000G           JSR     ADRDEV
8318    BD      0000G           JSR     BFRALC
831E    96      00G             LDA A   ERRCD,D
8320    26      30              BNE     ALFAIL
8322    C6      01              LDA B   1,I             ;SET UP FOR FIND
8324    D7      00G     ALLP:   STA B   R7,D
8326    BD      0000G           JSR     MTAFIN
8329    96      00G             LDA A   ERRCD,D         ;DID IT WORK
832B    26      25              BNE     ALFAIL          ;IF FAILED GO BACK TO CALLER
832D    96      00G             LDA A   MTSTAT,D
832F    85      1C              BIT A   28.,I
8331    27      1F              BEQ     ALFAIL
8333    88      0A              EOR A   10.,I
8335    85      0A              BIT A   10.,I
8337    27      05              BEQ     ALOKAY
8339    D6      00G     ALSKP:  LDA B   R7,D            ;TRY NEXT FILE
833B    5C                      INC B
833C    20      E6              BRA     ALLP
                ;
833E    BD      0000G   ALOKAY: JSR     UNADR
8341    86      80              LDA A   RUNFLG,I        ;SET RUN MODE
8343    9A      00G             ORA A   GLBFLG,D
8345    97      00G             STA A   GLBFLG,D
8347    86      00G             LDA A   OLDCOD,I
8349    97      00G             STA A   CTKN,D
834B    86      23              LDA A   35.,I
834D    97      00G             STA A   A.PRIM,D
834F    7E      0000G           JMP     OLDONE          ;FAKE ENTRY POINT INTO OLD
                ;
8352    7F      0000G   ALFAIL: CLR     ERRCD           ;FORGET ANY ERRORS
8355    7E      0000G           JMP     UNADR
                ;
        0001'                   END
```

```
TYPIN  KEYBOARD INPUT EDITOR  RT-11 MMAC  VM02-10  14-OCT-76 01:42:08 PAGE 14+
SYMBOL TABLE

ADRDEV= XXXXXX G    AFPITT= XXXXXX G    ALFAIL  8352      ALLP    8324      ALOKAY  833E
ALSKP   8339        AN    =  00F6       ANCOM   8296      ANCRMT= XXXXXX G  ANFLA   8253
ANFLB   8269        ANFNDA  8285        ANFNDJ  8302      ANHOLD= XXXXXX G  ANINCR= XXXXXX G
ANLPA   82A1        ANLPC   82D0        ANLPD   82EF      ANNEWL  82E7      ANSET   8249
ANSKA   82B0        ANSKC   82C7        ANSKPA  8266      ANSKPB  826C      ANSPAC  82FD
ANSTOP  82B1        ANYCHR  819F        ANYCLP  81A3      ATLD  =  00F7     ATLOAD  8300 G
ATLOD =  0008       AUTONO  8279 G      A.END =  XXXXXX G   A.MAX = XXXXXX G   A.PRIM= XXXXXX G
A.PTR =  XXXXXX G   A.STAT= XXXXXX G    BACKUP  80F1      BCKSCN  8100      BCKUPA  80FB
BDLT  =  0081       BFRALC= XXXXXX G    BFRSTT=  0020     BLOT  =  0080     BS    =  0008
BSCNA   8116        BSCNB   8122        BSCNLP  8104      BSCNSK  8110      BSTP  =  00F1
CLPTR =  XXXXXX G   CLRLN =  00F3       CLRLP   8200      CMPLN =  0080     CMPLNL  818C
CMPLNM  8192        CR    =  000D       CRCONT  8088      CRLOOP  807A      CRMOVE  8089
CRNORM=  0001       CRNULL  80A9        CRSKIP  8085      CRSTAT= XXXXXX G  CTKN  = XXXXX G
CTLCHR= XXXXXX G    DELETE  80E0        DOCR    8136      DSPCHR= XXXXXX G  EDTBFR= XXXXXX G
EDTCLR  81FB G      EDTCLS  822F G      EDTEND= XXXXXX G  EDTKEY  806A      EDTMAX= XXXXXX G
EDTPTR= XXXXXX G    ENDKEY=  0040       ERRCD = XXXXXX G  ERROR   81F7      EXPLN =  00F0
EXPLNL  8176        EXPLNM  8170        FDLT  =  0082     FIND  = XXXXXX G  FIX1  = XXXXXX G
FPC   = XXXXXX G    FSTP  =  00F2       FSTPSK  80C2      GLBFLG= XXXXXX G  GNMBR   82 3
INAGTE= XXXXXX G    INITMT= XXXXXX G    INLPA   8038      INLPB   804A      INPUTK=  0001
INSKPB  8064        INSKPC  8065        KBFLAG= XXXXXX G  MTAFIN= XXXXXX G  MTSASC= XXXXXX G
MTSPGM= XXXXXX G    MTSREG= XXXXXX G    MTSTAT= XXXXXX G  NEWCHR  8042      NLPTR = XXXXXX G
NMBS  =  0088       NMCR  =  008D       NPRC  =  0084     OLDCOD= XXXXXX G  OLDONE= XXXXXX G
OPRADR= XXXXXX G    PGMFP =  0003       PGMLNN=  0007     PGMPTR= XXXXXX G  PNMBR   8216
PRC   =  00F4       PRCFND  81E9        PRCLPA  81D9      PRCOK   81CD      PRCSKA  81E5
PRINT   814C        PRINTJ  8164        PRINTL  8154      PRINTN  814F      RCLN  =  0083
RTSA    8064        RTSB    8124        RTSB2   8182      RTSC    822E      RTSTWO  8180
RUBOUT=  007F       RUNFLG=  0080       R0    = XXXXXX G  R1    = XXXXXX G  R10   = XXXXXX G
R11   = XXXXXX G    R15   = XXXXXX G    R16   = XXXXXX G  R2    = XXXXXX G  R3    = XXXXXX G
R4    = XXXXXX G    R5    = XXXXXX G    R6    = XXXXXX G  R7    = XXXXXX G  R8    = XXXXXX G
R9    = XXXXXX G    SCRTCH= XXXXXX G    SPACE =  0020     TAN     8238      TATLD   8125
TBDLT   80C5        TBSTP   8001        TCLRLN  8132      TCMPLN  818A      TCR     8070
TEXPLN  8174        TFDLT   8080        TFSTP   808C      TITBL   8000      TITBLX=  8033
TMPHX = XXXXXX G    TNPRC   8183        TPRC    81C3      TRCLN   813B      TYPIN   8033 G
UNADR = XXXXXX G    UNCOMP= XXXXXX G    XAXIS = XXXXXX G  ZX    = XXXXXX G
  ABS.  8358    00
        0000    01
ERRORS DETECTED:  0     WARNINGS POSTED:  0     FREE CORE:  2756. WORDS
.SY:TYPIN/C(OK1:SEICLI,TYPIN
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

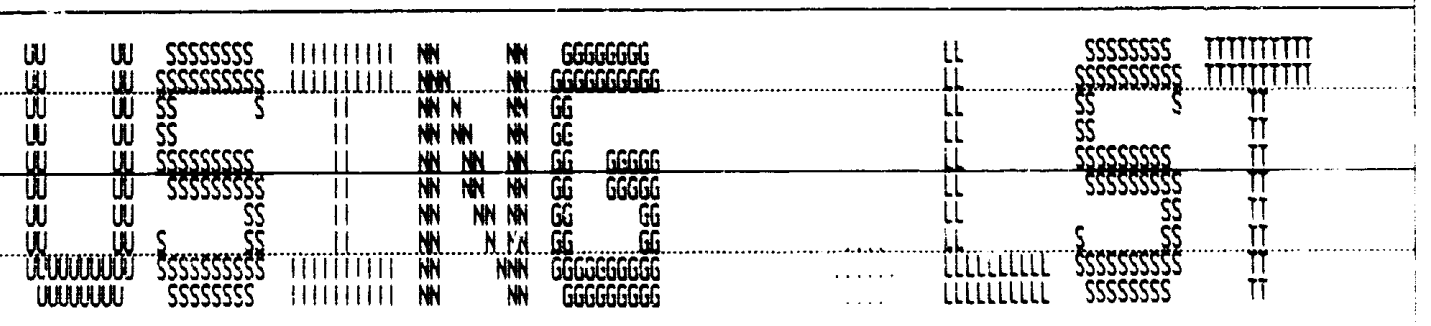
CROSS REFERENCE TABLE (CREF V01-03.)

XAXIS 1-46# 10-21 13-21
ZX 1-59# 7-14 10-37 11-38 12-9 13-9

CROSS REFERENCE TABLE (CREF V01-03)

SEI 1-3#

TBL 2-1# 2-10 2-11 2-12 2-13 2-14 2-15 2-16 2-17 2-18 2-19 2-20 2-21 2-22
2-23 2-24 2-25 2-26



```
                                .TITLE  USING   SET UP FORMAT CONTROLS ON STACK
                                .IDENT  /EB0006/
                                .GLOBL  USING
                                ;
                                .GLOBL  SYSERR,SETERA,SETERR
                                .GLOBL  GETLN,BACKUP
                                .GLOBL  A5X,A10X,D11X
                                ;
                                .GLOBL  ISP             ;ADDR OF FORMAT STRING
                                .GLOBL  ISL             ;BYTES OF FORMAT STRING DATA
                                .GLOBL  ISB             ;IMAGE STRING BASE (LINE OR NT ORIGIN)
                                ;
                                .GLOBL  CLPTR,VALTG,PLOSTG,PNTSTG,IMACOD
                                .GLOBL  ERRCD,ERLNNF
                                .GLOBL  R0,DREXTA
                                ;
0009                    PGMCD   =       9               ;DISP,D TO OBJECT CODE
0007                    NTWLEN  =       7               ;WORKING LEN STRING
0008                    NTSPTR  =       11.             ;ADDR OF DATA-5
                                ;
                                ;       SET UP POINTERS FOR IMAGE CONTROL STRINGS
                                ;
                                ;       INPUTS
                                ;       ENTRY ON STACK FOR IMAGE STRING
                                ;               LITERAL IF DATA IN PRINT LINE
                                ;               POINTER TO NAME TABLE IF STRING USED
                                ;               VALUE IF LINE NUMBER FOR IMAGE STATEMENT USED
                                ;
                                ;       OUTPUTS
                                ;       STACK IS PRUNED
                                ;       IMAGE CONTROL POINTERS ARE SET UP
                                ;
0000  30                USING:  TSX                     ;TEST TYPE OF LAST ENTRY ON STACK
0001  A6  00                   LDA A   0,X             ;GET TAG OFF STACK
0003  81  00G                  CMP A   VALTG,I         ;VALUE
0005  27  08                   BEQ     USNVAL
0007  81  00G                  CMP A   PLOSTG,I        ;STRING
0009  27  20                   BEQ     USNLIT
000B  81  00G                  CMP A   PNTSTG,I        ;STRING VARIABLE
000D  27  38                   BEQ     USNSTR
000F  BD  0000G                JSR     SYSERR          ;SOMEONE FAILED
                                ;
0012  BD  0000G         USNVAL: JSR     GETLN           ;CONVERT VALUE TO LINE PTR
0015  96  00G                  LDA A   ERRCD,D         ;DID IT WORK
0017  26  18                   BNE     USNERR
0019  DE  00G                  LDX     R0,D            ;MAKE SURE IT IS IMAGE STMT
001B  A6  09                   LDA A   PGMCD,X
001D  81  00G                  CMP A   IMACOD,I
001F  26  13                   BNE     USNERR
0021  FF  0000G                STX     ISB             ;FOR ON FULL
0024  BD  0000G                JSR     A10X            ;GET ADDR OF FORMAT DATA
0027  DF  00G                  STX     ISP,D           ;SAVE POINTERS FOR PRINT
0029  DE  00G                  LDX     R0,D            ;GET FORMAT DATA LEN IN BYTES
002B  EE  00                   LDX     0,X
002D  BD  0000G                JSR     D11X            ;DON'T COUNT HEADER
0030  DF  00G                  STX     ISL,D
0032  20  39                   BRA     IMAGE           ;GO JOIN COMMON CODE
```

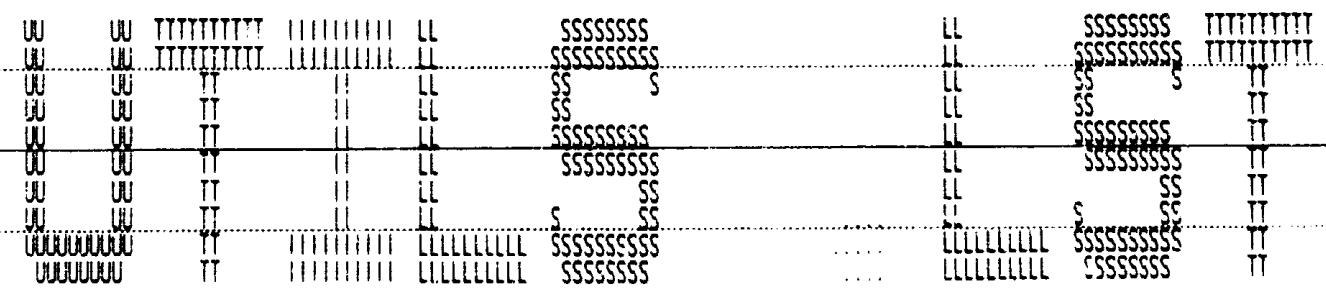
```
                                ;
0034    80  0000G   USNERR: JSR     SETERR
0037    00G                 .BYTE   ERLNNF
                            ;
0038    EE    03    USNLIT: LDX     3,X       ;X IS STILL SP
003A    E6    00            LDX     0,X
003C    DF    00G           STX     ISL,D     ;IT IS PTR TO COUNT
003E    DE    00G           LDX     CLPTR,D   ;PRINT IS ACTIVE LINE
0040    FF  0000G           STX     ISP
0043    30                  TSX               ;GET DATA LEN
0044    EE    03            LDX     3,X
0046    08                  INX               ;SKIP COUNT
0047    08                  INX
0048    20    1A            BRA     USNCOM
                            ;
004A    EE    01    USNSTR: LDX     1,X       ;X IS SP
004C    EE    07            LDX     NTWLEN,X  ;SPACE FOR DATA
004E    DF    00G           STX     ISL,D
0050    30                  TSX
0051    A6    01            LDA A   1,X       ;SAVE ANY WORK NT PTR
0053    8A    80            ORA A   128.,I
0055    B7  0000G           STA A   ISB
0058    A6    02            LDA A   2,X
005A    B7  0001G           STA A   ISB+1
005D    EE    01            LDX     1,X
005F    EE    08            LDX     NTSPTR,X  ;ADDR OF DATA
0061    8D  0000G           JSR     AEX
0064    DF    00G   USNCOM: STX     ISP,D
0066    9F    00G           STS     R0,D      ;BACK UP ONE ENTRY
0068    8D  0000G           JSR     BACKUP
0068    9E    00G           LDS     R0,D
                            ;
0060    7E  0000G   IMAGE:  JMP     DREXTA
                            ;
                    0001'           .END
```

```
USING  SET UP FORMAT CONTROLS  RT-11 MMAC  V#02-10   14-OCT-76 01:42:31 PAGE 1+
SYMBOL TABLE

A10X  = XXXXXX G      A5X   = XXXXXX G      BACKUP= XXXXXX G      CLPTR = XXXXXX G      DREXTA= XXXXXX G
D11X  = XXXXXX G      ERLNNE= XXXXXX G      ERRCD = XXXXXX G      GETLN = XXXXXX G      LNKCOD= XXXXXX G
IMAGE    0060R        ISB   = XXXXXX G      ISL   = XXXXXX G      ISP   = XXXXXX G      NTSPTR=   0008
NTWLEN=  0007         PGMCD =   0009        PLOSTG= XXXXXX G      PNTSTG= XXXXXX G      R0    = XXXXXX G
SETERA= XXXXXX G      SETERR= XXXXXX G      SYSERR= XXXXXX G      USING    0000RG       USNCOM   0064R
USNERR   0034R        USNLIT   0038R        USNSTR   004AR        USNVAL   0012R        VALTG = XXXXXX G

  ABS.    0000    00
          0070    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3376. WORDS
.SY:USING/C<DK1:SET.CLI.USING
```

CROSS REFERENCE TABLE (CREF V01-03.)

CROSS REFERENCE TABLE (CREF V01-0^)

SEL 1-3#



14-OCT-76

UTILS SYSTEM UTILITY ROUTINES RT-11 MACR VM02-10 14-OCT-76 01:42:38
TABLE OF CONTENTS

```
                                        .TITLE  UTILS   SYSTEM UTILITY ROUTINES
                                        .IDENT  /BEA025/
                                ;
                                        .GLOBL  UTILS
        0000'                   UTILS   =       .
                                ;
                                        .GLOBL  MSKCTR,ANYINT,TABPNT,PIATBL,SYSERR,IDLE
                                        .GLOBL  BANK,BANKSH,BNKADR
                                        .GLOBL  ITM1TG,RTRNTG,EOLTG,EOSTG
                                        .GLOBL  ERRCD,ERRCDB,CLOSE,SBP,PGMPTR,NLPTR,ZX,KEYSTK
                                        .GLOBL  R0,R1
                                ;
                                        .SBTTL  *** INTERRUPT CONTROLS
                                ;
                                ;       DISBLE - DISABLE ROM PACK INTERRUPTS
                                ;
                                ;       ENABLE - ENABLE ROM PACK INTERRUPTS
                                ;
                                ;       SAFE - SAFE TIME FOR ROM PACK INTERRUPT SERVICE
                                ;
                                        .GLOBL  ENABLE,DISBLE,SAFE
                                ;
        0000    B6      0000G   ENABLE: LDA A   MSKCTR          ;DON'T DEC RESET COUNTER
        0003    27      05              BEQ     SAFE            ;SEE IF INTERRUPTS NEED TO HAPEN
        0005    7A      0000G           DEC     MSKCTR          ;BACK OFF ONE LEVEL
        0008    26      18              BNE     INTRTS          ;IF NOT ZERO DON'T TRY FOR INTERRUPTS
                                ;
        000A    B6      0000G   SAFE:   LDA A   ANYINT          ;INTERRUPTS PENDING COUNT
        000D    27      13              BEQ     INTRTS          ;NONE
        000F    CE      0000G           LDX     PIATBL,I        ;SEARCH PIA TABLE
        0012    A6      02      INTLP:  LDA A   2,X             ;BIT 6 IS INTERRUPTS PENDING
        0014    85      40              BIT A   64.,I
        0016    26      0B              BNE     INTHIT
        0018    BD      00A9'           JSR     A6X             ;TRY NEXT ENTRY
        001B    A6      00              LDA A   0,X             ;LAST ENTRY HAS A ZERO ADDR
        001D    26      F3              BNE     INTLP
        001F    7F      0000G           CLR     ANYINT          ;IGNORE BAD COUNTER
        0022    39              INTRTS: RTS                     ;NONE SET - SOO LONG
                                ;
        0023                    INTHIT: SEI                     ;NO REAL INTERRUPTS NOW
0023    01      0F                      .BYTE   01,17
        0025    84      BF              AND A   255.-64.,I      ;RESET FLAG
        0027    A7      02              STA A   2,X
        0029    84      3F              AND A   63.,I           ;CLEAN UP BANK ADDR
        002B    BD      0041'           JSR     SETBNK          ;SET THE BANK SWITCH CTRL REG
        002E    EE      03              LDX     3,X             ;GET SERVICE ROUTINE ADDR
                                ;*****************************************************
                                ;       JSR     6,X             ;CALL HIS ROUTINE
        0030    AD      03              JSR     3,X             ;CALL CORRECT ENTRY ADDRESS
                                ;*****************************************************
        0032    BD      0040'           JSR     CLRBNK          ;RESETBANK SWITCH REG
        0035    0E                      CLI                     ;REAL INTERRUPTS ARE OKAY NOW
        0036    20      D2              BRA     SAFE            ;SEE IF THERE ARE MORE
                                ;
        0038    7C      0000G   DISBLE: INC     MSKCTR          ;BUMP MASK COUNT
        003B    26      E5              BNE     INTRTS
        003D    BD      0000G           JSR     SYSERR          ;IF COUNTER OVERFLOWS KILL SYSTEM
```

```
                                        .SBTTL  XXX BANK SWITCH CONTROL FUNCTIONS
                                ;
                                ;       SETBNK SET THE BANK SWITCH TO THE VALUE IN ACC-A.
                                ;       CLRBNK RESETS THE BANK SWITCH TO THE VALUE OF ZERO.
                                ;
                                        .GLOBL  SETBNK,CLRBNK
                                ;
0040    4F              CLRBNK: CLR A
0041    16              SETBNK: TAB
0042    07                      TPA
0043    36                      PSH A
0044                            SEI
0044    01      0F      .BYTE   01,17
0046    96      00G             LDA A   BANK,D
0048    84      C0              AND A   192.,I
004A    18                      ABA
004B    B7      0000G           STA A   BANKSW
004E    97      00G             STA A   BANK,D
0050    32                      PUL A
0051    06                      TAP
0052    39                      RTS
```

```
                                        .SBTTL  XXX SPECIAL EXECUTION STACK
                                ;
                                ;       PSHRET IS USED TO MOVE ONE RETURN ADDRESS TO SPECAIL STACK.
                                ;       RTRN IS USED TO GO BACK TO THE SAVED RETURN POINT.
                                ;
                                        .GLOBL  PSHRET,RTRN
                                        .GLOBL  XEQSP,DREXTB
                                ;
0053  DE    00G     PSHRET: LDX     XEQSP,D         ;GET STACK POINTER
0055  33                    PUL B                   ;SAVE CALLERS RETURN ADDR
0056  D7    01G             STA B   DREXTB+1,D
0058  33                    PUL B
0059  D7    02G             STA B   DREXTB+2,D
005B  33                    PUL B                   ;NOW STACK NEW RETURN POINT
005C  E7    00              STA B   0,X
005E  33                    PUL B
005F  E7    01              STA B   1,X
0061  08                    INX
0062  08                    INX
0063  DF    00G             STX     XEQSP,D
0065  30                    TSX
0066  7E    0000G           JMP     DREXTB
                                ;
0069  DE    00G     RTRN:   LDX     XEQSP,D
006B  09                    DEX
006C  09                    DEX
006D  DF    00G             STX     XEQSP,D
006F  EE    00              LDX     0,X
0071  16                    TAB
0072  6E    00              JMP     0,X
```

```
                                        .SBTTL  *** PUSH X REG

                                        .GLOBL  PUSHX,PUSHXT,R21

                                ;PUT INDEX TAG BY USER OR AS ITEM 1

0074    86      00G     PUSHX:  LDAA    #ITM1TG,1
0076    DF      00G     PUSHXT: STX     R21,D
0078    BD      0053'           JSR     PSHRET
007B    D6      01G             LDAB    R21+1,D
007D    37                      PSHB
007E    D6      00G             LDAB    R21,D
0080    37                      PSHB
0081    36                      PSHA
0082    7E      0069'           JMP     RTRN
```

XXX PULL ANY PSUEDO REGISTER

```
                                .SBTTL  XXX PULL ANY PSUEDO REGISTER
                                .GLOBL  PULLRW,R0,STAX
                                ;
                                ;       PULL TWO BYTES OFF THE STACK
                                ;
                                ;       BYTE AFTER HAS REG#2
                                ;
0085    30              PULLRW: TSX                     ;GET CALLERS INPUT
0086    EE      00              LDX     0,X
0088    DF      00G             STX     R21,0           ;SAVE IT FOR LATER
008A    A6      00              LDA A   1,X             ;GET REG NUMBER
008C    CE      0000G           LDX     R0,I            ;REGISTER FILE ADDRESS
008F    33                      PUL B                   ;FORGET PC ON STACK
0090    33                      PUL B
0091    33                      PUL B                   ;OLD TAG
0092    33                      PUL B                   ;HIGH BYTE
0093    BD      0000G           JSR     STAX
0096    4C                      INC A
0097    33                      PUL B
0098    BD      0000G           JSR     STAX
0098    DE      00G             LDX     R21,0           ;GET RETURN POINT
0090    6E      01              JMP     1,X
```

```
                                .SBTTL  XXX INDEX REG UPDATE ROUTINES
                        ;
                        ;       JUNK ROUTINES
                        ;
                                .GLOBL  A0X,A5X,A6X,A7X,A8X,A9X,A10X,A11X,A12X,A13X,A14X
                                .GLOBL  A15X,A16X
                        ;
009E    08      A16X:   INX
009F    08      A15X:   INX
00A0    08      A14X:   INX
00A1    08      A13X:   INX
00A2    08      A12X:   INX
00A3    08      A11X:   INX
00A4    08      A10X:   INX
00A5    08      A9X:    INX
00A6    08      A8X:    INX
00A7    08      A7X:    INX
00A8    08      A6X:    INX
00A9    08      A5X:    INX
00AA    08              INX
00AB    08              INX
00AC    08              INX
00AD    08              INX
00AE    08              INX
00AF    39      A0X:    RTS
                        ;
                                .GLOBL  D5X,D6X,D7X,D8X,D9X,D10X,D11X,D12X,D13X
                        ;
0080    09      D13X:   DEX
0081    09      D12X:   DEX
0082    09      D11X:   DEX
0083    09      D10X:   DEX
0084    09      D9X:    DEX
0085    09      D8X:    DEX
0086    09      D7X:    DEX
0087    09      D6X:    DEX
0088    09      D5X:    DEX
0089    09              DEX
008A    09              DEX
008B    09              DEX
008C    09              DEX
008D    39              RTS
```

```
                                        .SBTTL  XXX UPCASE FOR LEX AND ASCII TO FPN
                                        .GLOBL  UPCASE,GLBFLG,TESTCS
                0001            NCSFLG  =       1

                                ; THIS ROUTINE UPCASES ALL LOWERCASE ALPHA AND CONVERTS
                                ; BRAKETS AND BRACES TO PARENS; ALL OTHER CHARS ARE UNCHANGED


008E  36        TESTCS: PSH A
008F  96  006           LDA A   GLBFLG,D
0091  85  01            BIT A   #CSFLG,I
0093  32                PUL A
0094  26  0A            BNE     CSRTS
0096  81  60    UPCASE: CMP A   #'A,I
0098  23  14            BLS     NOTUP           ; LESS THAN "A"
009A  81  7A            CMP A   #'Z,I
009C  22  13            BHI     VERYUP          ; GREATER THAN "Z" -BOTH LOWERCASE
009E  84  DF            AND A   #%DF,I ; IF IT IS LOWERCASE UPCASE IT
00A0  39        CSRTS:  RTS                     ; AND QUIT
00A1  81  7D    VERYUP: CMP A   #'},I  ; IF ITS A BRACE CHANGE
00A3  26  02            BNE     1$              ; IT TO A PAREN
00A5  86  29            LDA A   #'),I
00A7  81  7B    1$:     CMP A   #'{,I
00A9  26  02            BNE     2$
00AB  86  28            LDA A   #'(,I
00AD  39        2$:     RTS
00AE  81  5B    NOTUP:  CMP A   #'[,I           ; IF ITS A BRAKET CHANGE
00B0  26  02            BNE     1$              ; IT TO A PAREN
00B2  86  28            LDA A   #'(,I
00B4  81  5D    1$:     CMP A   #'],I
00B6  26  02            BNE     2$
00B8  86  29            LDA A   #'),I
00BA  39        2$:     RTS
```

```
                                        .SBTTL  XXX RUNBNK      CALL ALL ROM PACKS
                                        .GLOBL  RUNBNK

                                        .GLOBL  R22,R23

                                ;       INPUTS
                                ;       X - ADDR OF ENTRY POINTS TO CALL

                                ;       OUTPUTS
                                ;       NONE

                                ;       R22 & R23 CAN'T BE USED BY ROM ROUTINES
                                ;
                                ;       BANK REGISTER FORMAT
                                ;
00EB  86  00G     RUNBNK: LDA A   RTRNTG,I        ;TAG RETURN POINT
00ED  36                  PSH A
00EE  07                  TPA                   ;SAVE CALLERS CC REG
00EF  97  01G             STA A   R23+1,D
00F1                      SEI                   ;RUN DISABLED
00F1  01  0F      .BYTE   01,17
00F3  DF  00G             STX     R22,D         ;SAVE ENTRY ADDR
00F5  4F                  CLR A                 ;PRIME BANK COUNTER
00F6  97  00G     BNKLP:  STA A   R23,D
00F8  BD  0041'           JSR     SETBNK
00FB  FE  0000G           LDX     BNKADR        ;GET ROM ID
00FE  8C  4051            CPX     16465.,I
0101  26  08              BNE     BNKSKP
0103  DE  00G             LDX     R22,D         ;ADDR OF ENTRY ADDR
0105  EE  00              LDX     0,X           ;ROUTINE ADDR
0107  27  02              BEQ     BNKSKP        ;NOT ONE
01C9  AD  00              JSR     0,X           ;GO TO BANK SETUP ROUTINE
0108  96  00G     BNKSKP: LDA A   R23,D         ;TRY NEXT ONE
010D  81  30              CMP A   48.,I         ;IF IN TOASTER SECTION GO BY ONES
010F  25  0D              BCS     BNKADD
0111  4C                  INC A
0112  81  40             CMP A   64.,I         ;LAST VALID BANK+1
0114  26  E0              BNE     BNKLP
0116  BD  0040'           JSR     CLRBNK
0119  96  01G             LDA A   R23+1,D
011B  06                  TAP
011C  32                  PUL A
011D  39                  RTS

011E  8B  08      BNKADD: ADD A   8.,I
0120  20  D4              BRA     BNKLP
```

```
                                        .SBTTL  XXX END         END STATEMENT
                                        .GLOBL  END
                                ;
                                ;       INPUTS
                                ;               NONE
                                ;
                                ;       OUTPUTS
                                ;               STACK IS RESET
                                ;
                                ;       NOTES
                                ;               RUN MODE IS RESET AND CONTROL IS PASSED TO THE IDLE LOOP
                                ;
0122    96      000     END:    LDA A   ERRCD,D         ;SAVE ERROR CODE
0124    97      000             STA A   ERRCDB,D
0126    86      000             LDA A   EOLTG,I         ;MAKE STACK LOOK CLEAN
0128    36                      PSH A
0129    BD      0000G           JSR     CLOSE           ;CLOSE ALL FILES
012C    96      000             LDA A   ERRCDB,D
012E    97      000             STA A   ERRCD,D
0130    7F      0000G           CLR     ERRCDB
0133                            SEI                     ;DISABLE INTERRUPTS FOR END
0133    01      0F              .BYTE   01,17
0135    9E      000             LDS     SBP,D           ;RESET STACK
0137    86      000             LDA A   EOSTG,I         ;TAG AS END OF STACK
0139    36                      PSH A
013A    DE      000             LDX     PGMPTR,D        ;WRAP AROUND PROGRAM
013C    DF      000             STX     NLPTR,D
013E    DE      000             LDX     ZX,D
0140    FF      0000G           STX     KEYSTK          ;RESET KEY STACK
0143    FF      0002G           STX     KEYSTK+2
0146    FF      0004G           STX     KEYSTK+4
0149    FF      0006G           STX     KEYSTK+6
014C    0E                      CLI                     ;ENABLE INTERRUPTS
014D    7E      0000G           JMP     IDLE            ;DIRTY EXIT TO IDLE LOOP
```

```
                                        .SBTTL  XXX OPRCL       GENERAL PURPOSE CALL FOR MATOPR AND EVAL
                                        .GLOBL  OPRCL
                                ;
                                ;       OPRCL - CALL FOR ALL COMMANDS, FUNCTIONS AND OPERATORS
                                ;       FROM EVAL (PGMEVL) AND MATOPR
                                ;
                                        .GLOBL  OPRADR,OPRRTN,XFNBNK,ERNXFN
                                        .GLOBL  LDAX,LDXX,STAX
                                ;
0150  DE   00G     OPRCL:       LDX     OPRADR,D       ;IF SIGN BIT SET IT IS DIRECT CALL
0152  2B   40                  BMI     OPRRDY
0154  32                       PUL A                  ;PULL RETURN ADDR
0155  B7   0000G              STA A   OPRRTN
0158  32                       PUL A
0159  B7   0001G              STA A   OPRRTN+1
015C  96   00G                LDA A   OPRADR,D       ;IF ZERO THIS IS BANK CALL
015E  27   08                 BEQ     BNKCL
0160  8A   80                 ORA A   128.,I
0162  97   01G                STA A   DREXTB+1,D
0164  96   01G                LDA A   OPRADR+1,D
0166  97   02G                STA A   DREXTB+2,D
0168  7E   0000G              JMP     DREXTB         ;GO TO IT (RETURN TO DREXTA)
                                ;
0168  96   01G     BNKCL:       LDA A   OPRADR+1,D     ;GET CONTROL BITS
016D  84   03                 AND A   3,I            ;GET JUXT BANK ADDR DISP
016F  CE   0000G              LDX     XFNBNK,I       ;GET BANK ADDR FROM POWER UP SEQUENCE
0172  BD   0000G              JSR     LDAX
0175  27   29                 BEQ     NOBNKB         ;TOO BAD
0177  BD   0041'              JSR     SETBNK
017A  FE   0000G              LDX     BNKADR         ;SEE IF USER GRABED IT
017D  8C   4051               CPX     16465.,I
0180  26   14                 BNE     NOBNKA
0182  96   01G                LDA A   OPRADR+1,D     ;GET VECTOR DISP
0184  44                       LSR A
0185  44                       LSR A
0186  FE   000AG              LDX     BNKADR+10.     ;GET VECTOR BASE
0189  BD   0000G              JSR     LDXX           ;GET ROUTINE ADDR
018C  AD   00                 JSR     0,X
018E  BD   0040'              JSR     CLRBNK         ;BE A HOME NOW
0191  FE   0000G     OPREXT:  LDX     OPRRTN         ;EXIT POINT
0194  6E   00        OPRRDY:  JMP     0,X
                                ;
0196  96   01G     NOBNKA:      LDA A   OPRADR+1,D     ;GET BANK AND RESET LINKAGE
0198  84   03                 AND A   3,I
019A  CE   0000G              LDX     XFNBNK,I
019D  5F                       CLR B
019E  BD   0000G              JSR     STAX
                                ;
01A1  86   00G     NOBNKB:      LDA A   ERNXFN,I
01A3  A7   00G                STA A   ERRCD,D
01A5  20   EA                 BRA     OPREXT
```

```
                                        .SBTTL  XXX DISKCL      DISK LINKAGE
                                        .GLOBL  DISKCL
                                        .GLOBL  JMPAX,ERNXFN,XFNBNK
                                ;
                                ;       ACC-A MUST HOME BRANCH CODE
                                ;       ACC-A=0 FOR TSTINT CALL
                                ;       OTHER CODES MUST BE MULT OF THREE
                                ;
01A2    36              DISKCL: PSH A                   ;SAVE CALLERS ENTRY CODE
01A3    B6      0002G           LDA A   XFNBNK+2
01A8    27      12              BEQ     DCOOPS          ;IF ZERO DISK IS NOT HOME
01AD    8D      0041'           JSR     SETBNK          ;SWITCH TO DISK ROM
01B0    FE      0000G           LDX     BNKADR          ;SEE IF IT IS STILL THERE
01B3    8C      4D61            CPX     #'MA+200        ;
01B6    26      07              BNE     DCOOPS
01B8    32                      PUL A                   ;GET CODE
01B9    FE      0000G           LDX     BNKADR+10       ;BASE FOR JUMP VECTOR
01BC    7E      0000G           JMP     JMPAX           ;HE WILL RTS
                                ;
01BF    32              DCOOPS: PUL A                   ;IF TSTINT CALL EXIT
01C0    4D                      TST A
01C1    27      04              BEQ     DCRTS
01C3    86      00G             LDA A   ERNXFN,I
01C5    97      00G             STA A   ERRCD,D
01C7    39              DCRTS:  RTS
                                ;
                0001'           .END
```

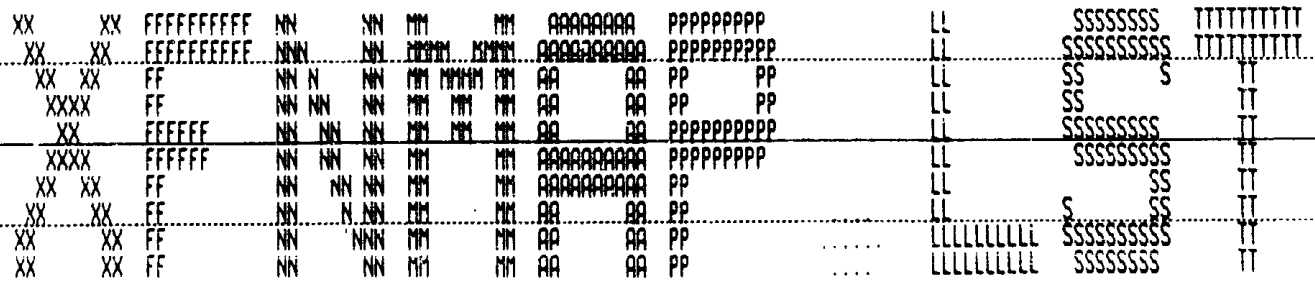
```
UTILS  SYSTEM UTILITY ROUTINES RT-11 MACRO  VM02-10  14-OCT-76 01:42:38 PAGE 11+
SYMBOL TABLE

ANY!NT= xxxxxx G    AOX     000FRG    A10X    000SRG    A11X    000HRG    A12X    000JRG
A13X    000KRG      A14X    000LRG    A15X    000MRG    A16X    000FRG    A5X     000BRG
A6X     000PRG      A7X     000RRG    A8X     000QRG    A9X     000NRG    BANK  = xxxxxx G
BANKSW= xxxxxx G    BNKADD  011ER     BNKADR= xxxxxx G    BNKCL   0168R     BNKLP   03F6R
BNKSKP  010BR       CLOSE = xxxxxx G  CLRBNK  004JRG    CSRTS   0000R     DCOOPS  018ER
DCRTS   01C7R       DISBLE  0038RG    DISKCL  01A7RG    DREXTB= xxxxxx G  D10X    0083RG
D11X    0082RG      D12X    0081RG    D13X    0080RG    D5X     0088RG    D6X     0087RG
D7X     0086RG      D8X     0085RG    D9X     0084RG    ENABLE  0000RG    END     0122RG
EOLTG = xxxxxx G    EOSTG = xxxxxx G  ERNXFN= xxxxxx G  ERRCD = xxxxxx G  ERRCD8= xxxxxx G
GLBFLG= xxxxxx G    IDLE  = xxxxxx G  INTHIT  0023R     INTLP   0012R     INTRTS  0022R
LTMTIG= xxxxxx G    JMPAX = xxxxxx G  KEYSTX= xxxxxx G  LDAX  = xxxxxx G  LDXX  = xxxxxx G
MSKCTR= xxxxxx G    NCSFLG=  0001     NLPTR = xxxxxx G  NOBNKP  0196R     NOBNKB  01A1R
NOTUP   000ER       OPRADR= xxxxxx G  OPRCL   0150RG    OPREXT  0191R     OPRRDY  0194R
OPRRTN= xxxxxx G    PGMPTR= xxxxxx G  PIATBL= xxxxxx G  PSHRET  0063RG    PULLRN  0085RG
PUSHX   0074RG      PUSHXT  0076RG    RTRN    0069RG    RTRNTG= xxxxxx G  RUNBNK  00E8RG
R0    = xxxxxx G    R1    = xxxxxx G  R21   = xxxxxx G  R22   = xxxxxx G  R22   = xxxxxx G
SAVE    0003RG      SPR   = xxxxxx G  SETBNK  0041RG    STAX  = xxxxxx G  SYSERR= xxxxxx G
TABPNT= xxxxxx G    TESTCS  008ERG    UPCASE  00C6RG    UTILS =  0000RG   VERYUP  0001R
XEQSP = xxxxxx G    XFNBNK= xxxxxx G  ZX    = xxxxxx G
  ABS.    0000    00
          01C8    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3092  WORDS
.SY:UTILS/C:DK1:SELCLI,UTILS
```

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

CROSS REFERENCE TABLE (CREF V01-03)

SE! 1-3# 1-55 2-12 8-20 9-21



```
                                .TITLE  XFNMAP  MAP FOR ORIGIN OF EXTENDER FUNCTION ROM
                                .IDENT  /BEB013/
                                ;
                                ;       OVER IS SET TO ONE FOR OVERLAY VERSION
                                ;
0000                    OVER    =       0
                                ;
                                .IF     EQ,OVER
8800                    BNKADR  =       H8800
                                .IFF
                        BNKADR  =       H5000
                                .ENDC
                                ;
                                .GLOBL  XFNMAP,BNKADR,BNKEND
0000                            .ASECT
8800                            . = BNKADR
8800                    XFNMAP  =       .
                                ;
                                .GLOBL  GRFINI,BANK,XFNBNK
                                ;
8800    4051                    .WORD   16465.          ;ID FOR REAL BANK
8802    8840                    .WORD   PWRUP           ;ENTRY FOR POWER UP
8804    0000G                   .WORD   GRFINI          ;AT INIT TIME
8806    0000G                   .WORD   GRFINI          ;DELETE ALL IS ALSO INIT
8808    0000                    .WORD   0               ;NO CLOSE ROUTINE
880A    880E                    .WORD   TRVCTR          ;BASE FOR SPECIAL VECTORS
880C      00                    .BYTE   0               ;BANK DYNAMIC MEMORY ID
880D      00                    .BYTE   0               ;END OF NAME LIST
                                ;
                                ;
                                .MACRO  TRV     NAME
                                .GLOBL  Q'NAME
                                .WORD   Q'NAME
                                .ENDM
                                ;
880E                    TRVCTR: TRV     AXIS            ;TRANSFER VECTORS FOR XFN ROM
8810                            TRV     SCALE
8812                            TRV     VIEW
8814                            TRV     WINDO
8816                            TRV     ROTAT
8818                            TRV     DRAW
881A                            TRV     RDRAW
881C                            TRV     MOVE
881E                            TRV     RMOVE
8820                            TRV     CROSS
8822                            TRV     GIN
8824                            TRV     ASC
8826                            TRV     CHR
8828                            TRV     LEN
882A                            TRV     POS
882C                            TRV     REP
882E                            TRV     SEG
8830                            TRV     CAT
8832                            TRV     VAL
8834                            TRV     STR
8836                            TRV     ASN
8838                            TRV     ACS
```

```
                                        TRV     ATN
                                        TRV     RND
                                        TRV     SETFUZ
                                ;
                                ;
                                ;
8840    96      00G     PWRUP:  LDA A   BANK.D          ;TELL I AM HERE
8842    B7      00000G          STA A   XENBNK          ;I AM FIRST ENTRY
8845    7E      00000G          JMP     GRFINI          ;AS PART OF POWER UP
                                ;
        8848                    BNKEND  =               ;END OF BASE DATA IN BANK
                                ;
                                ;
        0001'                   .END
```

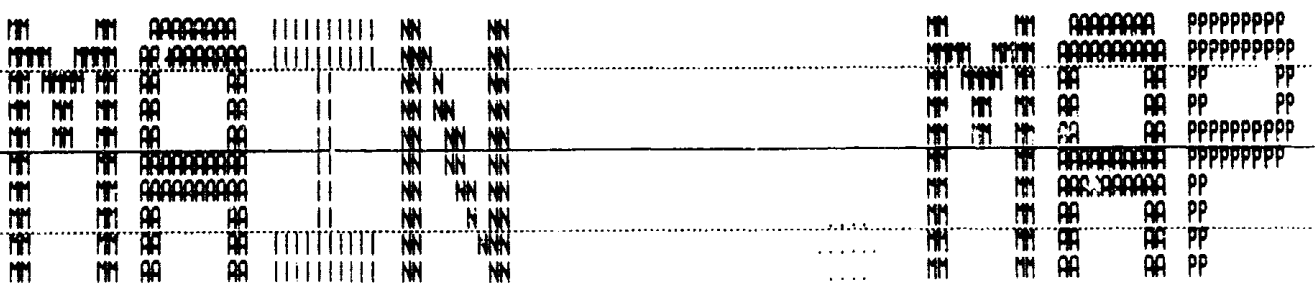
```
XFNMAP MAP FOR ORIGIN OF EXTEN RT-11 MMAC  VM02-10  14-OCT-76 01:42:52 PAGE 1+
SYMBOL TABLE

BANK  = XXXXXX G      BNKADR=  8800 G      BNKEND=  8848 G      GRFINI= XXXXXX G      OVER  =  0000
PWRUP    8840         QACS  = XXXXXX G      QASC  = XXXXXX G      QASN  = XXXXXX G      QATN  = XXXXXX G
QAXIS = XXXXXX G      QCAT  = XXXXXX G      QCHR  = XXXXXX G      QCROSS= XXXXXX G      QDRAW = XXXXXX G
QGIN  = XXXXXX G      QLEN  = XXXXXX G      QMOVE = XXXXXX G      QPOS  = XXXXXX G      QRDRAW= XXXXXX G
QREP  = XXXXXX G      QRMOVE= XXXXXX G      QRND  = XXXXXX G      QROTAT= XXXXXX G      QSCALE= XXXXXX G
QSEG  = XXXXXX G      QSTTFU= XXXXXX G      QSTR  = XXXXXX G      QVAL  = XXXXXX G      QVIEW = XXXXXX G
QWINDO= XXXXXX G      TRVCTR   880E         XFNBNK= XXXXXX G      XFNMAP=  8800 G
  ABS    8848    00
         0000    01
ERRORS DETECTED: 0     WARNINGS POSTED: 0     FREE CORE: 3321. WORDS
.SY:XFNMAP/C(DK1:SELCL1.XFNMAP
```

CROSS REF NCE TABLE (CREF V01-03)

	1-31#	1-32	1-83	
BANK	1-34#	1-79		
BNKADR	1-24#	1-29#	1-31	
BNKEND	1-29#	1-83#		
GRFINI	1-34#	1-38	1-39	1-81
OVER	1-21#	1-23		
PURUP	1-37	1-29#		
QACS	1-72	1-72#		
QASC	1-62	1-62#		
QASN	1-71	1-71#		
QATN	1-73	1-73#		
QAXIS	1-51	1-51#		
QCAT	1-68	1-68#		
QCHR	1-63	1-63#		
QCROSS	1-60	1-60#		
QDRAW	1-56	1-56#		
QGIN	1-61	1-61#		
QLEN	1-64	1-64#		
QMOVE	1-58	1-58#		
QPOS	1-65	1-65#		
QRDRAW	1-57	1-57#		
QREP	1-66	1-66#		
QRMOVE	1-59	1-59#		
QRND	1-74	1-74#		
QROTAT	1-55	1-55#		
QSCALE	1-52	1-52#		
QSEG	1-67	1-67#		
QSETFU	1-75	1-75#		
QSTR	1-70	1-70#		
QVAL	1-69	1-69#		
QVIEW	1-53	1-53#		
QWINDO	1-54	1-54#		
TRVCTR	1-41	1-51#		
XFNBNK	1-74#	1-80#		
XFNMAP	1-29#	1-32#		

CROSS REFERENCE TABLE (CREF V01-03)

SE1	1-3#												
TRV	1-46#	1-51	1-52	1-53	1-54	1-55	1-56	1-57	1-58	1-59	1-60	1-61	1-62	1-63
	1-64	1-65	1-66	1-67	1-68	1-69	1-70	1-71	1-72	1-73	1-74	1-75		



MULT DEF OF INTSRV
RT-11 CLNK V01-02 LOAD MAP
MAIN .LDA 19-OCT-76 17:27:06

SECTION	ADDR	SIZE	ENTRY	ADDR	ENTRY	ADDR	ENTRY	ADDR
ABS.	0000	8826	K0	0000	NULLTG	0000	R0	0000
			SLOPH	0000	AXIS	0000	FHEX	0000
			FUDGH	0001	K1	0001	PLOSTG	0001
			MOVCOD	0001	ERRPOW	0001	MTSSEC	0001
			IECDRV	0001	MTEOF	0001	ESTG	0002
			K2	0002	R1	0002	RMOCOD	0002
			ERRPDM	0002	MTSPGM	0002	MTSWRT	0002
			GOSTG	0003	K3	0003	PGFVCT	0003
			DRACOD	0003	ERUP	0003	FORTG	0004
			K4	0004	R2	0004	RDRCOD	0004
			EREXP	0004	MTSBIN	0004	FFOT	0004
			K5	0005	LISTTG	0005	SEGCOD	0005
			ERTRNG	0005	K6	0006	PGMTG	0006
			R3	0006	REPCOD	0006	ERSOR	0006
			SIZERR	0006	IMXTG	0007	POSCOD	0007
			ERLIMO	0007	PNTSTG	0008	R4	0008
			SCALE	0008	MINCOD	0008	CPYFLG	0008
			ERSHAP	0008	MTSASC	0008	FHIN	0008
			PSCTG	0009	MAXCOD	0009	ERDIM	0009
			PNTNTG	000A	R5	000A	ANDCOD	000A
			ERSUBC	000A	BOF	000A	PAETG	0008
			FLUNIT	000B	ORCOD	000B	ERNOEN	000B
			FRET	000B	R6	000C	VALTG	000C
			PLSCOD	000C	ERCLAI	000C	LNNOTG	000D
			MNSCOD	000D	ERUBYT	000D	FLEX	000D
			ITM1TG	000E	R7	000E	MULCOD	000E
			ERAPP1	000E	ITM2TG	000F	DIVCOD	000F
			ERAPP2	000F	LABKTG	0010	R8	0010
			VIEW	0010	XPCOD	0010	RWNDFG	0010
			ERFUZZ	0010	MTSNEW	0010	MTSRD	0010
			SEMITG	0011	DET	0011	EQUCOD	0011
			ERREN	0011	ATSNTG	0012	R9	0012
			NECOD	0012	ERMARK	0012	FSWP	0012
			BAKSTG	0013	LTCOD	0013	ERFORA	0013
			ERNXTA	0013	EROFA	0013	FLIN	0013
			ALLTG	0014	R10	0014	LTECOD	0014
			EREOFN	0014	RTRNTG	0015	GTECOD	0015
			ERASGN	0015	INT1	0016	PRTTG	0016
			R11	0016	GTCOD	0016	ERUPN	0016
			FDUP	0016	CALLTG	0017	CATCOD	0017
			ERLOG	0017	EOLTG	0018	INT2	0018
			R12	0018	WINDOW	0018	MPYCOD	0018
			ERARC	0018	FSAV	0018	EOSTG	0019
			INV	0019	SUMCOD	0019	ERSFE	0019
			FRES	0019	DIMCNT	001A	R13	001A
			TRNCOD	001A	ERCARC	001A	INVCOD	001A
			ERPOPN	001B	R14	001C	TPOINT	001C
			CHRCOD	001C	ERREP	001C	FART	001C
			STRCOD	001D	ERVAL	001D	FHIM	001D
			BYTCNT	001E	R15	001E	ASCCOD	001E
			ERMUL	001E	VALCOD	001F	ERDET	001F
			DIMLP	0020	R16	0020	ROTATE	0020
			LENCOD	0020	ERCNSF	0020	MTOPEN	0020
			TRN	0021	MPLCOD	0021	ERNCN	0021
			OSFLG	0022	R17	0022	MELCOD	0022
			ERNDT	0022	NOTCOD	0023	ERNIMX	0023
			RETH	0024	R18	0024	FNACOD	0024

ERUNDF	0024	RETL	0025	FNBCOD	0025
ERNXFN	0025	R19	0026	FNCCOD	0026
ERSEC	0026	FNDCOD	0027	ERWSFL	0027
R20	0028	DRAW	0028	FNECOD	0028
ERBRK	0028	MPY	0029	FNFCOD	0029
ERNON1	0029	R21	002A	FNGCOD	002A
ERNON2	002A	FNHCOD	002B	ERNON3	002B
R22	002C	FNICOD	002C	ERNON4	002C
FNJCOD	002D	ERNON5	002D	R23	002E
FNKCOD	002E	ERNON6	002E	FNLCOD	002F
ERNON7	002F	EQU	0030	RDRAW	0030
FNMCOD	0030	ERNON8	0030	CNT	0031
FNNCOD	0031	ERFBFR	0031	FLAG	0032
FNOCOD	0032	ERLOLD	0032	SHNTCT	0033
FNPCOD	0033	ERLNNF	0033	FNQCOD	0034
ERFNFD	0034	TFLGS1	0035	FNRCOD	0035
ERMTRD	0035	LEXCNT	0036	FNSCOD	0036
EREOM	0036	FNTCOD	0037	ERMILA	0037
SCORE	0038	MOVE	0038	FNUCOD	0038
ERWRT	0038	MASK	0039	FNVCOD	0039
ERNCRT	0039	COMP	003A	FNWCOD	003A
ERMFMT	003A	STPTR	003B	FNXCOD	003B
ERNOLD	003B	BRKCOD	003C	FNYCOD	003C
EROVRD	003C	PGMPTR	003D	FNZCOD	003D
ERMOPN	003D	TYPCOD	003E	ERFILE	003E
HEP	003F	ABSCOD	003F	ERREAD	003F
RMOVE	0040	SQRCOD	0040	ERSTGL	0040
MTFPOS	0040	ENDKEY	0040	ZX	0041
SIGCOD	0041	ERNSEP	0041	RNDCOD	0042
ERATSN	0042	SBP	0043	INTCOD	0043
ERNIOD	0043	LGNCOD	0044	EREXEC	0044
LSP	0045	LOGCOD	0045	ERIOE	0045
EXPCOD	0046	ERQUOT	0046	USRORG	0047
ASNCOD	0047	ERNFST	0047	CROSS	0048
ACSCOD	0048	ERINFS	0048	XEQSP	0049
ATNCOD	0049	ERIFC	0049	TANCOD	004A
ERINU	004A	ERRCD	004B	COSCOD	004B
ERTMDS	004B	ERRCDB	004C	SINCOD	004C
ERBPNU	004C	CLPTR	004D	PTCOD	004D
ERBMDU	004D	IECCOD	004E	ERLRSU	004E
NLPTR	004F	FILCOD	004F	ERBCMU	004F
GIN	0050	FORCOD	0050	ERBDPU	0050
NTPTR	0051	AS2COD	0051	ERDATM	0051
GOCOD	0052	ERTABQ	0052	CTKN	0053
GOSCOD	0053	ERDEOR	0053	LCLFLG	0054
IFCOD	0054	ERISGN	0054	GLBFLG	0055
RUNCOD	0055	ERISTS	0055	OPRADR	0056
RESCOD	0056	ERDAGN	0056	RENCOD	0057
ERFLDO	0057	DREXTA	0058	ASC	0058
DELCOD	0058	ERINTR	0058	APPCOD	0059
ERROM1	0059	LISCOD	005A	ERROM2	005A
DREXTB	0058	SAVCOD	005B	ERSTOP	005B
PRICOD	005C	ERTERM	005C	NEXCOD	005D
ERSYNX	005D	INTQQQ	005E	RETCOD	005E
ERCHAR	005E	STOCOD	005F	ERFIXN	005F
LOTEND	0060	CHR	0060	ENDCOD	0060
ERFXOV	0060	SETCOD	0061	EDTPTR	0062
INICOD	0062	GINCOD	0063	EDTMAX	0064
SLOPL	0064	POLCOD	0064	LETCOD	0065
ANCRNT	0066	MATCOD	0066	REACOD	0067
ANINCR	0068	LEN	0068	WRICOD	0068
INPCOD	0069	CRSTAT	006A	CALCOD	006A
KBFLAG	0068	WINCOD	0068	LSTKEY	006C
VIECOD	006C	PNDFLG	006D	AXICOD	006D

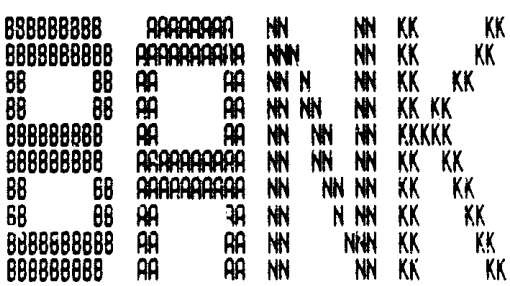
PNDEOF	006E	OLDCOD	006E	MAGEND	006F
CRECOD	006F	MTSTT2	0070	POS	0070
OPECOD	0070	DSPSTT	0071	KILCOD	0071
QIN	0072	SECCOD	0072	COPCOD	0073
QOUT	0074	ASSCOD	0074	UNICOD	0075
KBIN	0076	FINCOD	0076	PPMODE	0077
RBYCOD	0077	EOLCHR	0078	REP	0078
WBYCOD	0078	ETXCHR	0079	MARCOD	0079
NULCHR	007A	TLICOD	007A	STAT37	007B
PAGCOD	007B	BANK	007C	HOMCOD	007C
LUNNO	007D	DIRCOD	007D	RECNO	007E
CLOCOD	007E	WAICOD	007F	XAXIS	0080
SEG	0080	FUZCOD	0080	MTFFST	0080
MTFSIZ	0080	FA	0080	MTSREG	0081
SCACOD	0081	MTSTAT	0082	ROTCOD	0082
MTMAX	0083	CMDCOD	0083	CROCOD	0084
MTPTP	0085	ASCOD	0085	TOCOD	0086
NOOUT	0087	TO2COD	0087	TAGPTR	0088
CAT	0088	STECOD	0088	THECOD	0089
IOFLGS	008A	RELCOD	008A	CHRCNT	008B
OFCOD	008B	ASGCOD	008C	CHAR	008D
COLCOD	008D	IOFUNC	008E	SEMCOD	008E
A. STAT	008F	RPNCOD	008F	A. PRIM	0090
VAL	0090	COMCOD	0090	A. STRT	0091
USICOD	0091	LPNCOD	0092	A. MAX	0093
LBKCOD	0093	CRCOD	0094	A. END	0095
GRACOD	0095	DEGCOD	0096	A. SEC	0097
RADCOD	0097	A. PTR	0098	INPNT	0098
STR	0098	TRACOD	0098	NORCOD	0099
LENGTH	009A	KEYCOD	009A	NOKCOD	009B
POINT	009C	CASCOD	009C	NOCCOD	009D
COLCNT	009E	EOFCOD	009E	PICOD	009F
TABPTR	00A0	ASN	00A0	SPACOD	00A0
FS	00A0	MEMCOD	00A1	TCOL	00A2
DETCOD	00A2	IMACOD	00A3	YAXIS	00A4
REMCOD	00A4	TMP1	00A5	DATCOD	00A5
TMP2	00A6	DIMCOD	00A6	TMP3	00A7
DEFCOD	00A7	TMPHY	00A8	ACS	00A8
OFFCOD	00A8	TMPLY	00A9	ONCOD	00A9
TMPHX	00AA	SIZCOD	00AA	TMPLX	00AB
FULCOD	00AB	COLCT	00AC	SRQCOD	00AC
ROWCT	00AD	EOICOD	00AD	BLINK	00AE
ALLCOD	00AE	DOTON	00AF	SM2COD	00AF
FULL	00B0	ATN	00B0	NUMCOD	00B0
DISCNT	00B1	STGCOD	00B1	CURSOR	00B2
ITGCOD	00B2	FONT	00B3	LINCOD	00B3
MTXBYT	00B4	PRMCOD	00B4	RECCNT	00B5
LITCOD	00B5	CONCOD	00B6	LSTCOD	00B6
EARLYW	00B7	CCODES	00B8	RND	00B8
MTFLGS	00B9	BPIAMT	00BA	FILLOC	00BB
FILFNO	00BD	T1	00BF	T2	00C0
SETFUZ	00C0	FM	00C0	T3	00C1
T4	00C2	SIGNS	00C3	TABPNT	00C4
MULPNT	00C6	Y5	00C8	Y4	00C9
Y3	00CA	Y2	00CB	Y1	00CC
Y0	00CD	X5	00CE	X4	00CF
X3	00D0	X2	00D1	X1	00D2
X0	00D3	DEXP	00D4	EXM	00D4
EXL	00D5	ISP	00D6	DP	00D8
L1	00DA	SP1	00DB	L2	00DD
SP2	00DE	L3	00E0	FD	00E0
SP3	00E1	L4	00E3	SP4	00E4
CURSER	00E6	PARCNT	00E8	ISL	00E9
DL	00EB	DT	00ED	LITFLG	00EE

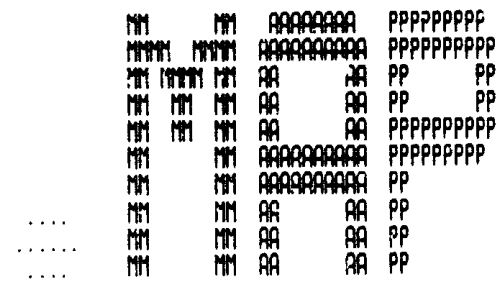
LRSFLG	00EF	NUMFLG	00F0	FFLG	00F1
PFLG	00F2	MFLG	00F3	DOLFLG	00F4
FUDGL	00F4	COMFLG	00F5	FMTFLG	00F6
FLAGS	00F7	IOSNTP	00F8	PGZERO	00FA
KEYQUE	0100	QEND	011E	TRASH	011E
MTBFR	011F	SAVE	0121	EDTBFR	0221
IOBFR1	0268	SCRTCH	0285	OPEN	0300
HCTRCS	0388	ONTBL	0389	EOFTBL	03C9
FNTBL	0300	KEYSTK	0411	CRASH	041A
ENDTBL	041A	XEQSTK	0424	HLDEOF	0434
ERCTRA	0435	ERCTRB	0436	OPRRTN	0437
COSPTR	0439	COOPTR	0438	BRKCNT	0430
JMPX	043E	LDXX	0445	LDAX	044B
LDBX	0451	STAX	0457	JMPAX	0450
ANHOLD	0462	IECRLF	0463	PPEOR	0464
PPEOF	0465	PPNUL	0466	IOSYSF	0467
MTMASK	0468	XFNBNK	0469	DSPVCT	0460
IOLVCT	0470	CHRVCT	0473	POINTB	0476
EDTFLG	0478	SCMSED	0479	XMINW	047A
XMAXW	0482	YMINW	048A	YMAXW	0492
DLTXS	049A	OLD	04A1	XMINS	04A2
DLTYS	04AA	YMINS	04B2	XSF	04BA
XLAST	04C2	YSF	04CA	YLAST	04D2
SINTHA	04DA	COSTHA	04E2	XNEW	04EA
TEMPX	04F2	YNEW	04FA	CREATE	0500
TEMPY	0502	MSKCTR	0508	ANYINT	0508
PIATBL	050C	PIAEND	058A	PGCTR	058C
FULSTT	058E	LINELN	058F	MTERR	0590
CNTL	0591	TTY2	0592	SHIFT1	0593
SHIFT2	0594	ROW	0595	COL	0596
STKUSE	0597	KEYTIM	0598	KEYCNT	0599
KBMRK	059A	KBSTK	059C	ACCEL	05A4
KYRATE	05A5	VLOKEY	05A6	DIGFLG	05A7
ITSINT	05A8	EFLAG	05A9	FPA	05AA
FPB	05B2	FPC	05BA	FUZA	05C2
FUZB	05C3	FUZR	05CB	CONV1	05CC
DETADD	05CE	KERNEL	05D6	ISB	05DE
DB	05E0	ES	05E2	E1	05E3
E2	05E4	E3	05E5	N1	05E6
N11	05F0	N12	05F1	NS	05F2
DOP	05F3	DN1	05F4	DN2	05F5
DIGCNT	05F6	F1	05F7	F2	05F8
EXP	05F9	EXS	05FA	LSPS	05FB
LDIG	05FC	LDIGA	05FD	LCOM	05FF
ZERO1	0600	ZERO2	0601	RDIG	0602
RDIGA	0603	ZERO3	0605	ZERCNT	0606
ZERSVH	0607	ZERSVL	0608	COMCNT	0609
DPCONT	060A	TABCNT	060B	PGNEND	060C
KILTYP	0721	UNIT	0800	DIR	0900
COFTYP	0A20	RELBL	0B00	PRINT	0C20
INPUT	009F	READ	0EA2	WRITE	0F21
ASSIGN	1000	SPCIOF	1020	LIST	1320
ZDRAW	1420	ZMOVE	1520	PAGE	1620
HOME	1720	ZGIN	1BA0	CMD	1A00
FIND	1B21	MARK	1C21	SECFNC	1D25
TYPIN	8033	EDTCLR	81FB	EDTCLS	822F
AUTONO	8279	ATLOAD	8300	PRTERR	8388
BUG16A	8746	IOPAT1	8752	CLOSP1	875A
FIX22	8766	WBYP3	876C	PIAHY	878C
PIALY	878E	PIAHX	8794	PIALX	8796
PIAMTA	8798	PIAMTB	879A	PIAKB	87A8
PIALT	87AA	PIAE01	87B0	PIASRQ	87B2
BANKSW	87C0	ACIA	87C6	BNKADR	8800
BNKMAP	8800	BNKEND	8816	ZRDRAW	9420

```
                                ZRMOVE  9520    TRAPS   FFF2
        8826    0000
        8826    0000
        8826    0000
LEX     8826    0721    Q0      887C    Q1      88E7    Q2      8CA2
                        Q3      8D8E    JUNK    8F18
OPTABL  8F48    015E
        9006    0000
MCD601  9006    0000
INFO    9006    0011
        9088    0086    GETLIN  9088    GETSMA  90F8    GETLAR  9106
        913E    00C6    PGMLIS  913E
        9204    0164    APPEND  9204
        9368    0161    EVLEN   9368    FNEVL   9368    QUSRFN  9380
                        DEF     9396    TSTINT  93C4
        94CA    0189    MATOPR  94CA    SNGMAT  94CA    MATMAT  9408
                        MATSCL  94E6    SCLMAT  94F4    LITREL  9598
                        SUM     9620
        9654    0255    PGMEVL  9683    LNEVL   968C
        98AA    02E5    UNCOMP  98AA
        9890    01C9    UNLEX   9890    PUTCHR  9C6E    NUMOUT  9043
        905A    016C    RENUMB  905A    RENUM   90C8
        9F16    0000
        9F16    01C8    FLCTRL  9F16    RUN     9F16    STOP    9F4E
                        GOTO    9F52    GOSUB   9F60    OF      9F87
                        RETURN  9FE4    IF      A013    FOR     A026
                        NEXT    A066
        A00E    0000
MCD302  A00E    0000
SHUNT   A00E    0128
        A206    063A    PARSE   A206    INCXN   A2CC    IMMTBL  A6A9
                        RELTBL  A73C
        A840    0070    USING   A840
        A880    00E3    TRANSL  A880
        A994    01C8    ENABLE  A994    UTILS   A994    SAFE    A99E
                        DISBLE  A9CC    CLRBNK  A9D4    SETBNK  A9D5
                        PSHRET  A9E7    RTRN    A9F0    PUSHX   AA08
                        PUSHXT  AA0A    PULLRN  AA19    A16X    AA33
                        A15X    AA34    A14X    AA35    A13X    AA36
                        A12X    AA37    A11X    AA38    A10X    AA39
                        A9X     AA3A    A8X     AA3B    A7X     AA3C
                        A6X     AA3D    A5X     AA3E    A0X     AA43
                        D13X    AA44    D12X    AA45    D11X    AA46
                        D10X    AA47    D9X     AA48    D8X     AA49
                        D7X     AA4A    D6X     AA4B    D5X     AA4C
                        TESTCS  AA52    UPCASE  AA5A    RUNBNK  AA7F
                        END     AA86    OPRCL   AAE4    DISKCL  AB3B
        AB5C    02C6    IOPROC  AB5C    IOCLNR  AB67    IOSCAN  AB86
                        SPCIOB  ABB3
        AE22    0000
IOCTL   AE22    0284    ADRDEV  AE22    BFRALC  AEC7    UNADR   AF1D
                        UNADR3  AF50    UNADR2  AF5E    PT      AF7F
                        AS      AF82    IEC     AF82    ATPROC  AF94
                        INAITT  B004    FPAITT  B014    AFPITT  B050
                        FIXNUM  B08F    INAGTE  B0C2
        B0D6    0030
MCD208  B0D6    0000
SYMTAB  B0D6    00C8
        B1A2    0000
FPMATH  B1A2    0900    FIX1    B1A2    FLOAT1  B1FB    FLOAT2  B1FB
                        SHMR    B210    FPSUB   B23E    FPADD   B245
                        NORM    B343    NORMR   B359    MAXANS  B3A1
                        UNDER   B3BC    ZANS    B3BC    ZIX     B3C2
                        ZIX4    B3CA    FPDIV   B3D3    FPMUL   B4AE
```

```
                        ASCFPN  B7?F    PSHFPN  B6EB    PULFPN  B70F
                        CMPFPN  B7C0    FUZZIE  B800    FPNASC  B853
                        RNDATA  B949    PLTAB   B902    NGTAB   BA3A
        BAA2    0000
EVLTBL  BAA2    013C    OPRTBL  BAA2    CMDTBL  BB1A
OCTBL   BBDE    004F
        BC2E    0227    SWIERR  BC2E    SYSERR  BC30    PWRUP   BC48
                        HIRAM2  BC66
        BE56    0256    DATACN  BE56    FLDATA  BE56    LITCN   BE69
                        INTCN   BE84    LINCN   BE84    PTRNTN  BE99
                        PTRNTS  BEB1    FNPARM  BEC1    FNASGN  BEE1
                        GTDATA  BF02    RESTZ   BF5C    RESTS   BF65
                        MATMOV  BF88    ASG     C003    ASSCSC  C030
        C0AC    0188    RSTMT   C0AC    EOF     C0AC    ONUNIT  C100
                        OFF     C107    ON      C11B    EQL     C12C
                        INIT    C143    INITX   C146    SRQOFF  C19F
                        DLTALL  C1A6    SPACE   C1D5    MEMORY  C1FB
                        RAD     C221    DEG     C228    GRAD    C22D
                        TRACE   C232    NORMAL  C239    KEY     C240
                        NOKEY   C244    CASE    C248    NOCASE  C24C
                        WAIT    C250
        C264    0000
        C264    0000
OUTCTL  C264    0279    PRISTG  C264    CRLFLF  C276    CRLF    C278
                        PRIVAL  C27E    PRIDFT  C288    PUTBYT  C330
                        OUTBFR  C374    SNDBFR  C404    NIODEV  C42C
                        XFRCTL  C431    CPSSET  C479    CPSIEC  C490
                        SCRMS   C4A9    SCRMO   C4CD
        C4DE    01D5    KBINT   C4DE    MTPRWD  C506    KBPROC  C590
                        IDLE    C562    QTRNSL  C62A    KBQOUT  C64A
                        CIDLE   C65C    KBQIN   C69A
        C684    05A2    CRTDRV  C684    DRBUSY  C684    DSPAGE  C68E
                        DSHOME  C6C8    WAIT.   C6E6    BELCAL  C727
                        PCHARZ  C7C5    CHSCAN  C888    VECTOR  C881
                        CRTRST  C8BF    PCHAR   C8EE
        CC56    0000
        CC56    021A    BINCTL  CC56    REAVAL  CCAA    REASTG  CCCF
                        WRISTG  CD3D    WRIVAL  CD6F    DATIN   CD9E
                        TYPE    CE00
        CE70    0307    EVLSUB  CE70    SETARG  CE70    CLRARG  CE74
                        TYPARG  CE77    TYPRES  CF77    MATSIZ  CFFA
                        PUSHES  D062    POPES   D07C    GETLNA  D082
                        GETLN   D087    LOCTG   D0E9    LOCTGR  D104
                        BACKUP  D110    NOP     D158    HALTR   D159
                        HALTA   D160    SETERR  D165    SETERA  D16A
        D178    0848    FMTPNT  D178    PRINTF  D178    DEFPNT  D183
                        FMTINI  D1D7    FMTCLN  D203
        D9C0    08C5    TAPE    D9C0    KILL    D9DA    APPOLD  D9EC
                        MODT88  D9FF    MODMTH  DA02    TYPNEW  DA1E
                        TYPBIN  DA26    TYPASC  DA2E    TYPLST  DA36
                        PRGBLN  DA3E    PRGPGM  DA48    PRGDAT  DA52
                        TYPSEC  DA5C    MTKILL  DA64    MTMARK  DAA0
                        MKFILE  DB0A    MTFIND  DBD1    MTAFIN  DBDD
                        MTREAD  DCA8    MTRDER  DD05    MTNULL  DD0C
                        MTCLOS  DD18    MTWRIT  DD3F    MTPOUT  DD98
                        MTBINT  DDF5    MTBSWP  DDFE    MTCHKS  DE21
                        MTPINW  DE31    MTPINR  DE53    MTPIN   DF35
                        MTRBFR  DF43    MTSSET  DF88    MTSCRT  DFE1
                        MTPADR  DFE8    INITMT  E00C    XFRSCN  E070
                        OLDONC  E08C    TAPFIL  E128    TAPF2   E12F
                        TLIST   E207
        E286    00C3    CALL    E286    CALLBS  E286
        E34A    0204    DIM     E34A    DIMSUB  E34A    RPN     E354
                        DIMSTR  E4D4    BYTCAL  E5C4    INTMLC  E5F9
```

```
                INTMLA  E5FB
        E61E    0000    IOKONS  E61E
        E61E    0210    DELETE  E61E    DELET   E6A8
        E82E    0048    STKBLO  E82E    FILEIN  E868    FILEOT  E868
                        FILES   E86C    FIL     E870    TYPFIL  E874
                        FILERQ  E876
        E87A    073E    ADMAIN  E87A    DOFP    E8DE    PI      E918
                        PICON   E91E    FPZERO  E926    FPTWO   E92E
                        OVLN2   E936    LOG2    E93E    OVLTN   E946
                        TABAD   E94E    FPONE   E950    TAN15   E982
                        DEGCON  E98A    GRDCON  E9CA    RADCON  E90A
                        NE      E9EA    EQ      E9EE    GTE     E9F2
                        LTE     E9F6    LT      E9FA    GT      E9FE
                        TRIG    EA2C    FPNEG   E6AA    SLN     E8B6
                        TRUTH   E8CB    SQR     E8DF    ETOX    EC4D
                        LGN     ED27    LOG     ED27    SIG     EDE7
                        ABS     EE06    NOT     EE0E    AND     EE25
                        OR      EE25    FPCMP   EE3E    UP      EE6E
                        MIN     EF18    MAX     EF2C    INT     EF3C
                        SETRND  EFA3
        EFB8    0000
INTSRV  EFB8    003F
        EFF8    018F    INPCTL  EFF8    INPSTG  EFF8    INPVAL  F033
                        GETCHR  F085    NEWBFR  F0B1    DSPGIN  F11F
                        KEYIN   F163
        F188    00A6    WBYTE   F188    WBYF1   F18A    RBYTE   F18F
                        WBYTVL  F1E2    RBYTVL  F20F
        F22E    022D    CTLCHR  F22E    DSPCHR  F2A6    FULSCN  F2C2
                        DSPOUT  F2EA    DSPCPY  F348    DSPCP2  F354
                        GENCUR  F36F    DSMOVE  F3A8    DSDRAW  F3AE
                        DSGRAF  F381    GSCNUM  F423    DSFONT  F42B
                        DSFULL  F434    DSPLY   F447
        F45C    0053    EXEC    F45C
        F4B0    02F9    INTACP  F4B0    IECOFF  F4D8    IECRD   F4E3
                        IECSND  F50E    INTSRC  F539    IECIFC  F55A
                        EOION   F560    EOIOFF  F576    ATNON   F57F
                        ATNOFF  F58A    IECEOI  F591    SRQRDY  F5AC
                        IECOUT  F5C0    IECIN   F60F    POLL    F6A2
                        SPOLON  F798
        F7AA    0031    CLOSE   F7AA    COPY    F7C8
        F7DC    01CD    GETKEY  F7DC    KEYDRV  F7DC
        F9AA    01E5    COMPR   F9AA    BUG16B  FACE    BUG16C  FB01
        FB90    0467    MTDRV   FB90    PUPTAP  FA90    REWIND  FBD7
                        CNFRCS  FC67    MARKS   FC73    WRRS    FD25
                        READRS  FD54    NMISRV  FDBC    WREND   FE4E
                        BAKARS  FE81    SKIPS   FE84    DELY1S  FEB0
                        DELY3S  FEB2    SERCHS  FECC    TST8NF  FF8C

UNDEFINED GLOBALS:
XFNMAP
STRING
GRAPH
GRAF
ADBANK

TRANSFER ADDRESS = 0001
HIGH LIMIT = FFF8
```





19-OCT-76

MULT DEF OF INTSRV
RT-11 CLIX V01-02 LOAD MAP
BANK .LDH 19-OCT-76 17:27:54

SECTION	ADDR	SIZE	ENTRY	ADDR	ENTRY	ADDR	ENTRY	ADDR
ABS.	0000	884E	K0	0000	NULLTG	0000	R0	0000
			SLOPH	0000	F[illegible]X	0000	FUDGH	0001
			K1	0001	P[illegible]	0001	MOVCOD	0001
			ERFPDV	0001	M[illegible]	0001	IECDRV	0001
			MTEOF	0001	ESTG	0002	K2	0002
			R1	0002	RMOCOD	0002	ERFPDV	0002
			MTSPGM	0002	MTSWRT	0002	GOSTG	0003
			K3	0003	DRACOD	0003	ERUP	0003
			FORTG	0004	K4	0004	R2	0004
			RDRCOD	0004	EREXP	0004	MTSBIN	0004
			FFOT	0004	K5	0005	LISTTG	0005
			SEGCOD	0005	ERTRNG	0005	K6	0006
			PGMTG	0006	R3	0006	REPCOD	0006
			EPSOR	0006	SIZERR	0006	IMXTG	0007
			POSCOD	0007	ERLNNO	0007	PNTSTG	0008
			R4	0008	MINCOD	0008	CPYFLG	0008
			ERSHAP	0008	MTSASC	0008	FHIN	0008
			PSCTG	0009	MAXCOD	0009	ERDIM	0009
			PNTNTG	000A	R5	000A	ANDCOD	000A
			ERSUBC	000A	BOF	000A	PAETG	000B
			ORCOD	000B	ERNOFN	000B	FRET	000B
			R6	000C	VALTG	000C	PLSCOD	000C
			ERCLAI	000C	LNNOTG	000D	MNSCOD	000D
			ERWBYT	000D	FLEX	000D	ITM1TG	000E
			R7	000E	MULCOD	000E	ERAPP1	000E
			ITM2TG	000F	DIVCOD	000F	ERAPP2	000F
			LBRKTG	0010	R8	0010	STRING	0010
			UPCOD	0010	RUNDFG	0010	ERFUZZ	0010
			MTSNEW	0010	MTSRD	0010	SEMITG	0011
			EQUCOD	0011	ERREN	0011	ATSNTG	0012
			R9	0012	NECOD	0012	ERMARK	0012
			FSWP	0012	BAKSTG	0013	LTCOD	0013
			ERFORA	0013	ERNXTA	0013	EROFA	0013
			FLIN	0013	ALLTG	0014	R10	0014
			LTECOD	0014	EREOFN	0014	RTRNTG	0015
			GTECOD	0015	ERASGN	0015	INT1	0016
			PRTTG	0016	R11	0016	GTCOD	0016
			ERUPN	0016	FDUP	0016	CALLTG	0017
			CATCOD	0017	ERLOG	0017	EOLTG	0018
			INT2	0018	R12	0018	MPYCOD	0018
			ERARC	0018	FSAV	0018	EOSTG	0019
			SUMCOD	0019	ERSFE	0019	FRES	0019
			DIMCNT	001A	R13	001A	TRNCOD	001A
			ERCARC	001A	INVCOD	001B	ERDOMN	001B
			R14	001C	TPOINT	001C	CHRCOD	001C
			ERREP	001C	FART	001C	STRCOD	001D
			ERVAL	001D	FHIM	001D	BYTCNT	001E
			R15	001E	ASCCOD	001E	ERMUL	001E
			VALCOD	001F	ERDET	001F	DIMLP	0020
			R16	0020	LENCOD	0020	ERCNSF	0020
			MTOPEN	0020	MPLCOD	0021	ERNCN	0021
			DSFLG	0022	R17	0022	MMICOD	0022
			ERNDT	0022	NOTCOD	0023	ERNIMX	0023
			RETH	0024	R18	0024	FNACOD	0024
			ERUNDF	0024	RETL	0025	FNBCOD	0025
			ERNXFN	0025	R19	0026	FNCCOD	0026
			ERSEC	0026	FNDCOD	0027	ERWSFL	0027

R20	0028	FNECOD	0028	ERBRK	0028
FNFCOD	0029	ERNON1	0029	R21	002A
FNGCOD	002A	ERNON2	002A	FNHCOD	002B
ERNON3	002B	R22	002C	FNICOD	002C
ERNON4	002C	FNJCOD	002D	ERNON5	002D
R23	002E	FNKCOD	002E	ERNON6	002E
FNLCOD	002F	ERNON7	002F	EQU	0030
FNMCOD	0030	ERNON8	0030	CNT	0031
FNNCOD	0031	ERFBFR	0031	FLAG	0032
FNOCOD	0032	ERLOLD	0032	SWNTCT	0033
FNPCOD	0033	ERLNWF	0033	FNQCOD	0034
ERFNFD	0034	TFLGS1	0035	FNRCOD	0035
ERMTRD	0035	LEXCNT	0036	FNSCOD	0036
EREOM	0036	FNTCOD	0037	ERMILA	0037
SCORE	0038	FNUCOD	0038	ERWRT	0038
MASK	0039	FNVCOD	0039	ERNCRT	0039
COMP	003A	FNWCOD	003A	ERMFMT	003A
STPTR	003B	FNXCOD	003B	ERNOLD	003B
BRKCOD	003C	FNYCOD	003C	EROVRD	003C
PGMPTR	003D	FNZCOD	003D	ERMOPN	003D
TYPCOD	003E	ERFILE	003E	HEP	003F
ABSCOD	003F	ERREAD	003F	SQRCOD	0040
ERSTGL	0040	MTFPOS	0040	ENDKEY	0040
ZX	0041	SIGCOD	0041	ERNSEP	0041
RNDCOD	0042	ERATSN	0042	SBP	0043
INTCOD	0043	ERHIOD	0043	LGNCOD	0044
EREXEC	0044	LSP	0045	LOGCOD	0045
ERIOE	0045	EXPCOD	0046	ERQUOT	0046
USRORG	0047	ASNCOD	0047	ERNFST	0047
ACSCOD	0048	ERINFS	0048	XEQSP	0049
ATNCOD	0049	ERIFC	0049	TANCOD	004A
ERINU	004A	ERRCD	0048	COSCOD	0048
ERTMDS	004B	ERRCDB	004C	SINCOD	004C
ERBPNU	004C	CLPTR	004D	PTCOD	004D
ERBMDU	004D	IECCOD	004E	ERLRSU	004E
NLPTR	004F	FILCOD	004F	ERBCMU	004F
FORCOD	0050	ERBDPU	0050	NTPTR	0051
AS2COD	0051	ERDATM	0051	GOCOD	0052
ERTABO	0052	CTKN	0053	GOSCOD	0053
ERDEOR	0053	LCLFLG	0054	IFCOD	0054
ERISGN	0054	GLBFLG	0055	RUNCOD	0055
ERISTS	0055	OPRADR	0056	RESCOD	0056
ERDAGN	0056	RENCOD	0057	ERFLDO	0057
DREXTA	0058	DELCOD	0058	ERINTP	0058
APPCOD	0059	ERROM1	0059	LISCOD	005A
ERROM2	005A	DREXTB	0058	SAVCOD	0058
ERSTOP	0058	PRICOD	005C	ERTERM	005C
NEXCOD	005D	ERSYNX	005D	INTQAQ	005E
RETCOD	005E	ERCHAR	005E	STOCOD	005F
ERFIXN	005F	EDTEND	0060	ENDCOD	0060
ERFXOV	0060	SETCOD	0061	EDTPTR	0062
INICOD	0062	GINCOD	0063	EDTMAX	0064
SLOPL	0064	POLCOD	0064	LETCOD	0065
ANCRNT	0066	MATCOD	0066	REACOD	0067
ANINCR	0068	WRICOD	0068	INPCOD	0069
CRSTAT	006A	CALCOD	006A	KBFLAG	006B
WINCOD	0068	LSTKEY	006C	VIECOD	006C
PNDFLG	006D	AXICOD	006D	PNDEOF	006E
OLDCOD	006E	MAGEND	006F	CRECOD	006F
MTSTT2	0070	OPECOD	0070	DSPSTT	0071
KILCOD	0071	QIN	0072	SECCOD	0072
COPCOD	0073	QOUT	0074	ASSCOD	0074
UNICOD	0075	KBIN	0076	FINCOD	0076
PPMODE	0077	RBYCOD	0077	EOLCHR	0078

ABYCOD	0078	ETXCHR	0079	MARCOD	0079
NULCHR	007A	TLICOD	007A	STATJ37	007B
PAGCOD	007B	BANK	007C	HOMCOD	007C
LUNNO	007D	DIRCOD	007D	RECNO	007E
CLOCOD	007E	WAICOD	007F	XAXIS	0080
FUZCOD	0080	MTFFST	0080	MTFSIZ	0080
FA	0080	MTSREG	0081	SCACOD	0081
MTSTAT	0082	ROTCOD	0082	MTMAX	0083
CMDCOD	0083	CROCOD	0084	MTPTR	0085
ASCOD	0085	TOCOD	0086	NOOUT	0087
TO2COD	0087	TABPTR	0088	STECOD	0088
THECOD	0089	IOFLGS	008A	RELCOD	008A
CHRCNT	008B	OFCOD	008B	ASGCOD	008C
CHAR	008D	COLCOD	008D	IOFUNC	008E
SEMCOD	008E	A.STAT	008F	RPNCOD	008F
A.PRIM	0090	COMCOD	0090	A.STRT	0091
USICOD	0091	LPNCOD	0092	A.MAX	0093
LBKCOD	0093	CRCOD	0094	A.END	0095
GRACOD	0095	DEGCOD	0096	A.SEC	0097
RADCOD	0097	A.PTR	0098	INPNT	0098
TRACOD	0098	NORCOD	0099	LENGTH	009A
KEYCOD	009A	NOKCOD	009B	POINT	009C
CASCOD	009C	NOCCOD	0090	COLCNT	009E
EOFCOD	009E	PICOD	009F	TABPTR	00A0
SPACOD	00A0	FS	00A0	MEMCOD	00A1
TCOL	00A2	DETCOD	00A2	IMACOD	00A3
YAXIS	00A4	REMCOD	00A4	TMP1	00A5
DATCOD	00A5	TMP2	00A6	DIMCOD	00A6
TMP3	00A7	DEFCOD	00A7	TMPHY	00A8
OFFCOD	00A8	TMPLY	00A9	ONCOD	00A9
TMPHX	00AA	SIZCOD	00AA	TMPLX	00AB
FULCOD	00AB	COLCT	00AC	SRQCOD	00AC
ROWCT	00AD	EOICOD	00AD	BLINK	0JAE
ALLCOD	00AE	DOTON	00AF	SM2COD	00AF
FULL	00B0	NUMCOD	00B0	DISCNT	00B1
STGCOD	00B1	CURSOR	00B2	ITGCOD	00B2
FONT	00B3	LINCOD	00B3	MTXBYT	00B4
PRMCOD	00B4	RECCNT	00B5	LITCOD	00B5
CONCOD	00B6	LSTCOD	00B6	EARLYW	0C97
CCODES	00B8	MTFLGS	00B9	BPIAMT	GJBA
FILLOC	00B8	FILFND	00BD	T1	00BF
T2	00C0	FM	00C0	T3	00C1
T4	00C2	SIGNS	00C3	TABPNT	00C4
MULPNT	00C6	Y5	00C8	Y4	00C9
Y3	00CA	Y2	00CB	Y1	00CC
Y0	00CD	X5	00CE	X4	00CF
X3	00D0	X2	00D1	X1	00D2
X0	00D3	DEXP	00D4	EXH	00D4
EXL	00D5	ISP	00D6	DP	00D8
L1	00DA	SP1	00DB	L2	00DD
SP2	00DE	L3	00E0	FD	00E0
SP3	00E1	L4	00E3	SP4	00E4
CURSER	00E6	PARCNT	00E8	ISL	00E9
DL	00EB	DT	00ED	LITFLG	00EE
LRSFLG	00EF	NUMFLG	00F0	FFLG	00F1
PFLG	00F2	MFLG	00F3	DOLFLG	00F4
FUDGL	00F4	COMFLG	00F5	FMTFLG	00F6
FLAGS	00F7	IOSNTP	00F8	PGZEND	00FA
KEYQUE	0100	QEND	011E	TRASH	011E
MTBFR	011F	SAVE	0121	EDTBFR	0221
IO9FR1	0268	SCRTCH	0285	OPEN	0300
HCTRCS	0388	ONTBL	0389	EOFTBL	03C9
FNTBL	0300	KEYSTK	0411	CRASH	041A
ENDTBL	041A	KEQSTK	0424	HLDEOF	0434

		ERCTRA	0435	ERCTRB	0436	OPRRTN	0437
		CDSPTR	0439	CDOPTR	043B	BRKCNT	043D
		JMPX	043E	LDXX	0445	LDAX	0448
		LDBX	0451	STAX	0457	JMPAX	0450
		ANHOLD	0462	IECRLF	0463	PPEOR	0464
		PPEOF	0465	PPNUL	0466	IOSYSF	0467
		MTMASK	0468	XFNBNK	0469	DSPVCT	046D
		IDLVCT	0470	CHRVCT	0473	POINTB	0476
		EDTFLG	0478	SCMSED	0479	XMINW	047A
		XMAXW	0482	YMINW	0484	YMAXW	0492
		DLTXS	049A	OLD	0491	XMINS	04A2
		DLTYS	04AA	YMINS	04B2	XSF	04BA
		XLAST	04C2	YSF	04CA	YLAST	04D2
		SINTHA	04DA	COSTHA	04E2	XNEW	04EA
		TEMPX	04F2	YNEW	04FA	CREATE	0500
		TEMPY	0502	MSKCTR	050A	ANYINT	0508
		PIATBL	050C	PIAEND	058A	PGCTR	058C
		FULSTT	058E	LINELN	058F	MTERR	0590
		CNTL	0591	TTY2	0592	SHIFT1	0593
		SHIFT2	0594	ROW	0595	COL	0596
		STKUSE	0597	KEYTIM	0598	KEYCNT	0599
		KBMRK	059A	KBSTK	059C	ACCEL	05A4
		KYRATE	05A5	VLDKEY	05A6	DIGFLG	05A7
		ITSINT	05A8	EFLAG	05A9	FPA	05AA
		FPB	05B2	FPC	05BA	FUZA	05C2
		FUZB	05C3	FUZR	05CB	CONV1	05CC
		DETADD	05CE	KERNEL	05D6	ISB	05DE
		D8	05E0	ES	05E2	E1	05E3
		E2	05E4	E3	05E5	N1	05E6
		N11	05F0	N12	05F1	NS	05F2
		DCP	05F3	DN1	05F4	DN2	05F5
		DIGCNT	05F6	F1	05F7	F2	05F8
		EXP	05F9	EXS	05FA	LSPS	05FB
		LDIG	05FC	LDIGA	05FD	LCOM	05FF
		ZERO1	0600	ZERO2	0601	RDIG	0602
		RDIGA	0603	ZERO3	0605	ZERCNT	0606
		ZERSVH	0607	ZERSVL	0608	COMCNT	0609
		DPCONT	060A	TABCNT	060B	PGNEND	060C
		KILTYP	0721	UNIT	0800	DIR	0900
		COPTYP	0A20	RELBL	0B00	PRINT	0C20
		INPUT	009F	READ	0EA2	WRITE	0F21
		ASSIGN	1000	SPCIOF	1020	LIST	1320
		ZDRAW	1420	ZMOVE	1520	PAGE	1620
		HOME	1720	ZGIN	18A0	CMD	1A00
		FIND	1B21	MARK	1C21	SECFNC	1D25
		TYPIN	8033	EDTCLR	81FB	EDTCLS	822F
		AUTONO	8279	ATLOAD	8300	PRTERR	8388
		BUG16A	8746	IOPAT1	8752	CLOSP1	875A
		FIX22	8766	WBYP3	876C	PIAHY	878C
		PIALY	878E	PIAHX	8794	PIALX	8796
		PIAMTA	8798	PIAMTB	879A	PIAKB	87A8
		PIALT	87AA	PIAEOI	87B0	PIASRQ	87B2
		BANKSW	87C0	ACIA	87C6	ANKADR	8800
		XFNMAP	8800	BNKEND	8848	ZRDRAW	9420
		ZRMOVE	9520	TRAPS	FFF8		
884E	0000						
884E	0000						
884E	02F7	QAXIS	884E				
8846	03CE	QVAL	8871	QSTR	88A0	QPOS	8BDC
		QSEG	8C84	QCAT	8D48	QREP	8DB8
		QLEN	8E86	QASC	8ECE	QCHR	8EE6
8F14	0000						
8F14	00E3	QCROSS	8F14	CROS	8F34		
8FF8	05E5	GRAF	8FF8	GRFINI	8FF8	QROTAT	904F

```
                                QWINDO  90C2    QVIEW   9126    QSCALE  916C
                                QDRAW   9208    QRDRAW  9212    QMOVE   9219
                                QRMOVE  9220    QGIN    9227    GRAPH   922E
                                GINSET  92E3    PLOTVL  9380    GRFDR.  945F
                                NEWSEC  9502
        950E    0222    ADBANK  950E    QACS    950E    QASN    950E
                        QATN    95FE    QRND    96F9    QSETFU  97A2
        9800    01C8    ENABLE  9800    UTILS   9800    SAFE    980A
                        DISBLE  9838    CLRBNK  9840    SETBNK  9841
                        PSHRET  9853    RTRN    9869    PUSHX   9874
                        PUSHXT  9876    PULLRN  9885    A16X    989F
                        A15X    98A0    A14X    98A1    A13X    98A2
                        A12X    98A3    A11X    98A4    A10X    98A5
                        A9X     98A6    A8X     98A7    A7X     98A8
                        A6X     98A9    A5X     98AA    A0X     98AF
                        D13X    98B0    D12X    98B1    D11X    98B2
                        D10X    98B3    D9X     98B4    D8X     98B5
                        D7X     98B6    D6X     98B7    D5X     98B8
                        TESTCS  98BE    UPCASE  98C6    RUNBNK  98EB
                        END     9922    OPRCL   9950    DISKCL  99A7
        99C8    02C6    IOPROC  99C8    IOCLNR  99D3    IOSCAN  99F2
                        SPCIOB  9A1F
        9C8E    0000
IOCTL   9C8E    0284    ADRDEV  9C8E    BFRALC  9D33    UNADR   9D89
                        UNADR3  9D8C    UNADR2  9DCA    PT      9DE8
                        PS      9DEE    IEC     9DEE    ATPROC  9E00
                        INAITT  9E70    FPAITT  9E80    AFPITT  9EC9
                        FIXNUM  9EF8    INAGTE  9F2E
        9F42    0000
MCD208  9F42    0000
SYMTAB  9F42    0078
        A00E    0000
FPMATH  A00E    0500    FIX1    A00E    FLOAT1  A067    FLOAT2  A067
                        SHMR    A089    FPSUB   A0AA    FPADD   A0B1
                        NORM    A1AF    NORMR   A1C5    MAXANS  A200
                        UNDER   A228    ZANS    A228    ZIX     A22E
                        ZIX4    A236    FPDIV   A23F    FPMUL   A31A
                        ASCFPN  A308    PSHFPN  A557    PULFPN  A578
                        CMPFPN  A62C    FUZZIE  A679    FPNASC  A68F
                        RNDATA  A785    PLTAB   A83E    NGTAB   A8A6
        A90E    0000
EVLTBL  A90E    013C    OPRTBL  A90E    CMDTBL  A986
OCTBL   AA4A    004F
        AA9A    0227    SWIERR  AA9A    SYSERR  AA9C    PWRUP   AAB7
                        HIRAMZ  AAD2
        ACC2    C256    DATACN  ACC2    FLOATA  ACC2    LITCN   ACD5
                        INTCN   ACF0    LINCN   ACFC    PTRNTN  AD05
                        PTRNTS  AD10    FNPARM  AD20    FNASGN  AD40
                        GTOATA  AD6E    RESTZ   D0C8    RESTS   AD01
                        MATMOV  ADF7    ASG     3E6F    ASSCSC  AE9C
        AF18    0188    BSTMT   AF18    EOF     AF18    ONUNIT  AF6C
                        OFF     AF73    ON      AF87    EOL     AF98
                        INIT    AFAF    INITX   AFB2    SRQOFF  B008
                        DLTALL  B012    SPACE   B041    MEMORY  B067
                        RAD     B08D    DEG     B094    GRAD    B099
                        TRACE   B09E    NORMAL  B0A5    KEY     B0AC
                        NOKEY   B0B0    CASE    B0B4    NOCASE  B0B8
                        WAIT    B0BC
        B0D0    0000
        B0D0    0000
OUTCTL  B0D0    0279    PRISTG  B0D0    CRLFLF  B0E2    CRLF    B0E4
                        PRIVAL  B0EA    PRIDFT  B0F4    PUTBYT  B1A9
                        OUTBFR  B1E0    SNDBFR  B270    NIODEV  B298
                        XFRCTL  B290    CPSSET  B2E5    CPSIEC  B309
```

```
                SCRMS   2?:S    SCRMO   B339
        B34A    01D5    KBINT   B34A    MTPRWD  B372    KBPROC  B3FC
                IDLE    B41E    QTRNSL  B496    KBQOUT  B4B6
                CIDLE   B4C8    KBQIN   B506
        B520    05A2    CRTDRV  B520    DRBUSY  B520    DSPAGE  B52A
                DSHOME  B534    WAIT.   B552    BELCAL  B593
                PCHARZ  B631    CHSCAN  B6F4    VECTOR  B9ED
                CRTRST  BA2B    PCHAR   BA5A
        BAC2    0000
        BAC2    021A    BINCTL  BAC2    REAVAL  BB16    REASTG  BB38
                WRISTG  BBA9    WRIVAL  BBD8    DATIN   BCDA
                TYPE    BC6C
        BCDC    0307    EVLSUB  BCDC    SETARG  BCDC    CLRARG  BCE0
                TYPARG  BCE3    TYPRES  BDE3    MATSIZ  BE66
                PUSHES  BECE    POPES   BEF8    GETLNA  BEEE
                GETLN   BEF3    LOCTG   BF55    LOCTGR  BF70
                BACKUP  BF89    NOP     BFC4    HALTR   BFC5
                HALTA   BFCC    SETERR  BFD1    SETERA  BFD6
        BFE4    0848    FMTPNT  BFE4    PRINTF  BFE4    DEFPNT  BFEF
                FMTINI  C043    FMTCLN  C074
        C82C    08C5    TAPE    C82C    KILL    C846    APPOLD  C858
                MOCT88  C868    MOOMTH  C86E    TYPNEW  C88A
                TYPBIN  C892    TYPASC  C89A    TYPLST  C8A2
                PRGBLN  C8AA    PRGPGM  C8B4    PRGDAT  C8BE
                TYPSEC  C8C8    MTKILL  C8D0    MTMARK  C90C
                MKFILE  C976    MTFIND  CA3D    MTAFIN  CA49
                MTREAD  CB14    MTRDER  CB71    MTNULL  CB78
                MTCLOS  CB84    MTWRIT  CBAB    MTPOUT  CC04
                MTBINT  CC61    MTBSWP  CC6A    MTCHKS  CC8D
                MTPINW  CC90    MTPINR  CCBF    MTPIN   CDA1
                MTRBFR  CDAF    MTSSET  CE27    MTSCRT  CE4D
                MTPADR  CE54    INITMT  CE78    XFRSCN  CEDC
                OLDONE  CEF8    TAPFIL  CF97    TAPF2   CF9B
                TLIST   D073
        D0F2    00C3    CALL    D0F2    CALLBS  D0F2
        D1B6    02D4    DIM     D1B6    DIMSUB  D1B6    RPN     D1C0
                DIMSTR  D340    BYTCAL  D430    INTMLC  D465
                INTMLA  D4[-
        D48A    0000    IOKONS  D48A
        D48A    0210    DELETE  D48A    DELET   D517
        D69A    0048    STKBLD  D69A    FILEIN  D6D4    FILEOT  D6D4
                FILES   D6D8    FIL     D6DC    TYPFIL  D6E0
                FILERQ  D6E2
        D6E6    073E    ADMAIN  D6E6    DOFP    D74A    PI      D784
                PICON   D78A    FPZERO  D792    FPTWO   D79A
                OVLN2   D7A2    LOG2    D7AA    OVLTN   D7B2
                TABAD   D7BA    FPONE   D7BC    TAN15   D81E
                DEGCON  D826    GRDCON  D836    RADCON  D846
                NE      D856    EQ      D85A    GTE     D85E
                LTE     D862    LT      D866    GT      D86A
                TRIG    D898    FPNEG   DA16    SLN     DA22
                TRUTH   DA37    SQR     DA48    ETOX    DA89
                LGN     D893    LOG     D893    SIG     DC53
                ABS     DC72    NOT     DC7A    AND     DC91
                OR      DC91    FPCMP   DCAA    UP      DCDA
                MIN     DD87    MAX     DD98    INT     DDA8
                SETRND  DE0F
        DE24    0000
INTSRV  DE24    003F
        DE64    018F    INPCTL  DE64    INPSTG  DE64    INPVAL  DE9F
                GETCHR  DEF1    NEWBFR  DF1D    DSPGIN  DF8B
                KEYIN   DFCF
        DFF4    00A6    WBYTE   DFF4    WBYP1   E026    RBYTE   E02B
                WBYTVL  E04E    RBYTVL  E078
```

E09A	022D	CTLCHR	E09A	DSPCHR	E112	FULSCN	E12E
		DSPOUT	E156	DSPCPY	E187	DSPCP2	E1C0
		GENCUR	E1D8	DSMOVE	E214	DSDRAW	E21A
		DSGRAF	E21D	GSCNUM	E28F	DSFONT	E297
		DSFULL	E2A0	DSPLY	E2B3		
E2C8	0053	EXEC	E2C9				
E31C	02F9	INTRCP	E31C	IECOFF	E344	IECRD	E34F
		IECSND	E37A	INTSRC	E3AE	IECIFC	E3C6
		EOION	E3D9	EOIOFF	E3E2	ATNON	E3EB
		ATNOFF	E3F6	IECEOI	E3FD	SRQRDY	E418
		IECOUT	E42C	IECIN	E478	POLL	E50E
		SPOLON	E604				
E616	0031	CLOSE	E616	COPY	E634		
E648	01C0	GETKEY	E648	KEYCHK	E648		
E816	01E5	COMPR	E816	PUG16B	E95A	BUG16C	E96D
E9FC	0467	MTDRV	E9FC	PUPTAP	E9FC	REWIND	EA43
		CNFRCS	E9D3	MARKS	EADF	WRRS	EB91
		READRS	EBC0	MMISRV	EC28	WREND.	ECBA
		BAKARS	ECED	SKIPS	ECF0	DELY1S	ED1C
		DELY3S	ED1E	SERCHS	ED38	TST8NF	EDF8

UNDEFINED GLOBALS:
GETSMA
GETLAR
TSTINT
WINDOW
VIEW
VAL
TRN
SUM
STR
STOP
SETFUZ
SEG
SCALE
RUN
ROTATE
RND
RMOVE
RETURN
REP
RDRAW
POS
OF
NEXT
MPY
MOVE
LEN
INV
IF
GOTO
GOSUB
GIN
FOR
FNEVL
FLUNIT
DRAW
DET
DEF
CHR
CAT
ATN
ASN

```
ASC
ACS
LNEVL
USING
UNLEX
UNCOMP
TRANSL
SHUNT
P_NUMB
PGMLIS
PGMEVL
PARSE
MATOPR
LEX
INFO
GETLIN
FLCTRL
EVLEN
CROSS
BNKMAP
AXIS
APPEND


TRANSFER ADDRESS = 0001
HIGH LIMIT = EE64
```