```
                INSERT  $IOMACS         TSSX,TSSX1,CALLIO,OPEN
TSSX    MACRO   R,X             USUAL SUPERVISOR-CALL MACRO
        TSX     $'R,X
        RMT
        IFF     $'R,NOISE
        TSSX1   R               DEFINE IT IF NOW UNDEFINED
        RMT
TSSX    END

TSSX1   MACRO   R
R       TIA     *+1
        BCI     1,R
TSSX1   END


CALLIO  MACRO   ROUTIN,FILE,ARGS
        STL     REDOIO          FOR RECOVERY PURPOSES
        TSSX    ROUTIN,4
        THREE   FILE,,
        THREE   FILE+1,,
        IRP     ARGS
        THREE   ARGS
        IRP
CALLIO  END

OPEN    MACRO   DRECTN,FILE
        STL     REDOIO
        TSSX    OPEN,4
        THREE   =H'DRECTN
        THREE   FILE,,
        THREE   FILE+1,,
        THREE   WRMODE,,        HARMLESS ON READING
        THREE   =2,,            DISK
        THREE   DISKER,,        ERROR RETURN
OPEN    END


ERROR   MACRO   LEGEND
        SXD     $ERROR,4
        TSX     $ERROR+1,4
        BCI     1,LEGEND
ERROR   END
```

```
                                        ABS
                          00024         ORG     20
00024  0500 00 0 00030                  CLA     AWD
00025  0601 00 0 77777                  STO     -1
00026  0020 00 0 00102                  TRA     XLDABS          FIX UP THE BUGS IN LDABS COMMAND
00027  0021 00 0 01625  FLAPCX  ITR     FLAPTR
00030  0 77743 0 77777  AWD     PZE     -1,,-STS
00031  0 77746 0 11227  NILSXX          $PNAME,,-*-1
00032  0 00000 0 77745                  -*-1
00033 -0 00000 0 77744                  MZE     -*-1
00034 -053143777777                     OCT     453143777777        NIL

00035  0 77742 0 13231   STS            $APVAL,,-*-1
00036 -0 77740 0 77741                  MZE     -*-1,,-*-2
00037  0 00000 0 00001                  1                       IS A CONSTANT ,,1 FOR APPLY
00040  0 77737 0 11227                  $PNAME,,-*-1
00041  0 00000 0 77736                  -*-1
00042 -0 00000 0 77735                  MZE     -*-1
00043 -146354777777                     OCT     546354777777
00044                    COMMON BSS     24
00074 +000000000012      Q10    DEC     10              CONSTANT MOVED TO GET COMPWD  SYN  00077
00075  0021 00 0 14552  FLAPCY ITR     C$LINK          ..
                                HEAD    B
00076  0 77747 0 77777  ZERC   PZE     -1,,-$NILSXX
                          00076 FLAPCZ  SYN     ZERC
00077  0 00000 0 00000  COMPWD PZE                     NONZERO FOR COMPILER PRINTOUT
00100  0 00000 0 00000  LISTNW PZE                     FILLED IN BY LAP TO STR
00101  0020 00 0 10145          TRA     S$EVQDN         TO LISTEN PROGRAM
00102  0074 00 4 01443  XLDABS TSX     $SETBK,4
00103  0520 00 0 01711          ZET     $LBT            SHOULD BE ZERO FOR NEW SYSTEM
00104  0020 00 0 10153          TRA     EVALQT          SYSTEM ALREADY SETUP
00105  0500 00 0 00076          CLA     ZERC            SYSTEM MUST BE SETUP
00106  0601 00 0 00000          STO     0
00107  0500 00 0 00027          CLA     FLAPCX           FOR FLOATING POINT TRAP
00110  0601 00 0 00010          STO     8
00111  0500 00 0 00075          CLA     FLAPCY           FOR COMPILER ROUTINE LINKAGE
00112  0601 00 0 00002          STO     2
00113  0500 00 0 00132          CLA     $AMASK           GET MEMORY BOUND OF 77777
00114                            TSSX    SETMEM,4
00115  0074 00 4 76676          TSX     E$SETUP,4       SETUP GC PARAMETERS
00116  0020 00 0 10153          TRA     EVALQT           GO READ INPUT
```

```
                                        HEAD    0
                                *       CONSTANT POOL

00117 +001000000000             Q01Q9   OCT     1000000000
00120 +233000000000             Q233Q9  OCT     233000000000
00121 -377000000000             Q777Q9  OCT     777000000000
00122   0 00000 1 00000         QT1             ,1
00123   0 00000 2 00000          QT2            ,2
00124   0 00000 4 00000          QT4            ,4
00125   0 00000 5 00000         QT5             0,5
00126   0 00001 0 00000          QD1    PZE     ,,1
00127   0 00007 0 00000          QD7    PZE     ,,7
00130  -1 00000 0 00000         QP5     STR
00131  -0 00000 0 00000          SBIT   MZE
00132   0 00000 0 77777          AMASK  PZE     -1
00133   0 77777 0 00000          DMASK  PZE     ,,-1
00134  -3 00000 0 00000         PMASK   TXL     0,0,0
00135  -3 77777 7 00000         PDTMSK  SVN     0,7,-1
00136 +000077000000              CNTMSK  OCT     000077000000
00137   0 00000 7 00000         TAGMSK  PZE     ,7
00140  -3 77777 7 77777         SEVENS  SVN     -1,7,-1
00141                           BCONAT  BSS     0               BEGINNING OF CONSTANT ATOMS
00141   0 13231 0 00000                 PZE     ,,APVAL
00142   0 12135 0 00000         FIXD    PZE     ,,BIN
00143   0 12116 0 00000         FLOATD          ,,$FLOAT
00144   0 12065 0 00000                 PZE     ,,FSUBR
00145   0 12057 0 00000         FNARGD  PZE     ,,FUNARG
00146   0 11723 0 00000         LABELD  PZE     ,,LABEL
00147   0 11712 0 00000         LAMDAD  PZE     ,,LAMBDA
00150   0 11334 0 00000          OCTD           ,,$OCT
00151   0 11227 0 00000         PNAMED  PZE     ,,PNAME
00152   0 11100 0 00000         QUOTED  PZE     ,,QUOTE
00153   0 10465 0 00000          SUBRD  PZE     ,,SUBR
00154   0 10540 0 00000         QSPECD  PZE     0,,SPECAL
00155   0 10446 0 00000          QSYMD  PZE     0,,SYM
00156   0 11625 0 12354                         ERSETO,,PJ36
00157   0 11603 0 11636                         PJ37,,PJ38          LOGAND LOGXCR
00160   0 11513 0 11524                         MAXP,,MINP
00161   0 10416 0 11251                         PLUS,,TIMES
00162   0 07644 0 07643                         H01,,H02            PROTECT INTEGER OBJECTS
00163   0 07646 0 07645                         H03,,H04
00164   0 07650 0 07647                         H05,,H06
00165   0 07652 0 07651                         H07,,H10
00166   0 00000 0 07642          H00A   PZE     H00
00167   0 00000 0 07654                 PZE     H12
00170   0 00000 0 07734          H72A   PZE     H72
00171   0 07653 0 00000          H11D   PZE     ,,H11
00172   0 07675 0 00000          H33D   PZE     ,,H33
00173   0 07676 0 00000          H34D   PZE     ,,H34
00174   0 07702 0 00000          H40D   PZE     ,,H40
00175   0 07736 0 00000          H74D   PZE     ,,H74
00176                           ECONAT  BSS     0               END OF CONSTANT ATOMS
00176                           WBUF    BSS     432
01056   0 00000 0 00000         OUTSW   PZE
```

```
                                *      ALL DISK ERRORS COME HERE, WHERE WE GO DORMANT IN SUCH A
                                *  CLEVER WAY THAT START WILL RETRY THE OFFENDING CALL.

01057                           DISKER  TSSX    WRFLX,4         THE USER EVEN GETS A BRIEF MESSAGE
01060   0 00010 0 01072                 TSX     DISKNG,,8       ..
01061                                   TSSX    DORMNT,4        HERE WE GO DORMANT
01062                                   TSSX    GETCOM,4        BACK TO EVALQUOTE IF HE SAID GIVEUP
01063   0 00000 0 00001                 TSX     1               ..
01064  -0340 00 0 15300                 LAS     =HGIVEUP        ..
01065   0020 00 0 01067                 TRA     REDOIO          .. IF HE DIDN'T SAY GIVEUP, RETRY
01066   0020 00 0 01070                 TRA     *+2             *HE SAID GIVEUP, WE SAY ERROR A1
01067   0020 00 0 00000 REDOIO          TRA     **              THIS WAS SET JUST BEFORE THE CALL
01070  -0754 00 0 00000                 ZAC                     ERROR WILL PRINT NIL
01071   0020 00 0 01670                 TRA     ERROR1

01072   243162426025    DISKNG  BCI     8,DISK ERROR.  SAVE AND PRNTER, OR START GIVEUP.

01102   606043316247    FILNAM  BCI     2,  LISP  LISP
01104   606043316247    OUTFIL  BCI     2,  LISPLSPOUT

                                * ROUTINE TO ASSIGN AN OUTPUT FILE

01106   0634 00 4 01134 ASIGN   SXA ASG,4
01107                           CALLIO  DELFIL,OUTFIL,TRAOPN
01114   0020 00 0 01116         TRA     TRAOPN
 1115   0634 00 4 01134 APEND   SXA     ASG,4
 1116                   TRAOPN  OPEN    W,OUTFIL
 1126                           CALLIO  BUFFER,OUTFIL,((WBUF,,432),DISKER)
01134   0774 00 4 00000 ASG     AXT     **,4
01135   0020 00 4 00001         TRA     1,4
                                *THE FOLLOWING ROUTINES CHANGE MODE OF OUTPUT FILE
                                *PUNCHT CAUSES ALL FILES TO BE WRITTEN IN TEMPORARY MODE
                                *PUNCHP CAUSES THEM TO BE PUNCHED IN PERMANENT MODE

01136   0500 00 0 15245 PUNCHT  CLA     =1
01137   0601 00 0 01144         STO     WRMODE
01140  -0754 00 0 00000         ZAC
01141   0020 00 4 00001         TRA     1,4
01142  -0754 00 0 00000 PUNCHP  ZAC                     MUST RETURN NIL AS WELL AS SET ZERO MODE
01143   0020 00 0 01137         TRA     PUNCHT+1

01144  +000000000000    WRMODE  DEC     0               INITIAL MODE IS PERMANENT

                                *BPSREMAINDER RETURNS THE NUMBER OF REMAINING WORDS IN BPS

01145   0634 00 4 01153 BPSREM  SXA     BPSA,4
01146  -0534 00 4 03022         LXD     C$LBPTP,4
01147   0754 00 4 00000         PXA     0,4
01150   0402 00 0 01704         SUB     ORG
01151   0560 00 0 12621         LDQ     Q$NVTBL+1
01152   0074 00 4 11163         TSX     MKNO,4
01153   0774 00 4 00000 BPSA    AXT     **,4
01154   0020 00 4 00001         TRA     1,4
```

```
                                *  ROUTINE TO SEEK A FILE FOR READING

01155  0634 00 4 01174  SEEK    SXA     SEEKRT,4
01156                    CALLIO  CLOSE,FILNAM,*+1 IGNORE PROBLEM OF FILE ALREADY OPEN
01163  0074 00 4 05452           TSX     TEREAD,4        INDICATE CALL WAS FOR DISK
01164                    OPEN    R,FILNAM        NOW OPEN THE FILE FOR READING
01174  0774 00 4 00000  SEEKRT  AXT     **,4
01175  0020 00 4 00001           TRA     1,4

                                *               VERBOSITY VALVES

01176  0601 00 0 02477  GCGAG   STO     VERBOS          GC CHECKS VERBOS FOR IO
01177  0020 00 4 00001           TRA     1,4
01200  0601 00 0 00077  COMPRI  STO     COMPWD          COMPILER SHOULD CHECK FOR IO
01201  0020 00 4 00001           TRA     1,4             TO DETERMINE WHETHER TO PRINT
```

```
                                        INSERT  $IOCNGF         FILE ROUTINES
01202  0634 00 4 01207  RDFLX   SXA     RDFLXX,4
01203 -0625 00 0 01215           STL     RDINDC          TELL READ THAT IO IS FROM TY
01204  0074 00 4 04755           TSX     $READ,4         IN THE READ ROUTINE
01205  0600 00 0 01215           STZ     RDINDC          CLEAR INDICATOR THAT IC IS FROM TY
01206 -0625 00 0 05276           STL     PRDIND          SAY PREVIOUS DEVICE READ WAS TY
01207  0774 00 4 00000  RDFLXX  AXT     **,4
01210  0020 00 4 00001           TRA     1,4
01211                    RDFLX1  TSSX    RDFLXA,4        ROUTINE TO READ TYPEWRITER
01212  0 00014 0 06363           PZE     I$CELL-12,,12   READ INTO END OF BUFFER
01213 -0625 00 0 05276           STL     PRDIND          SAY PREVIOUS DEVICE WAS TY
01214  0020 00 0 05372           TRA     I$GTEOF-2
01215  0 00000 0 00000  RDINDC  PZE                     INDICATOR FOR TY IC
                         *
                         *  ROUTINE TO OBTAIN THE PNAME OF AN OBJECT
                         *
01216  0634 00 4 01231  GETBCD  SXA     GETBA,4
01217  0560 00 0 00151           LDQ     PNAMED
01220  0074 00 4 10547           TSX     C$GET,4
01221  0074 00 4 13441           TSX     R$CARP,4
01222 -0734 00 4 00000           PDX     0,4
01223 -0500 00 4 00000           CAL     0,4
01224  0602 00 0 01240           SLW     GETBB
01225 -0320 00 0 15271  GETBG    ANA     =077            CHANGES TRAILING 77(OCTAL)
01226  0322 00 0 15271           ERA     =077            TO A 60(OCTAL)
01227  0100 00 0 01233           TZE     GETBD
01230 -0500 00 0 01240           CAL     GETBB
01231  0774 00 4 00000  GETBA   AXT     **,4
01232  0020 00 4 00001           TRA     1,4
01233 -0500 00 0 01240  GETBD    CAL     GETBB
01234  0771 00 0 00006           ARS     6
01235  0361 00 0 15304           ACL     =H 00000
01236  0602 00 0 01240           SLW     GETBB
01237  0020 00 0 01225           TRA     GETBG
01240  0 00000 0 00000  GETBB   PZE
                         *       THE GTFLNM ROUTINE IS CALLED WITH POINTERS IN THE AC AND MQ.
                         *  GTFLNM USES GETBCD TO DERIVE PRIMARY AND SECONDARY FILE NAMES FROM
                         *  THESE POINTERS, RESPECTIVELY.  THE RESULTS ARE LEFT IN FILNAM AND
                         *  FILNAM+1.
01241  0634 00 4 01250  GTFLNM  SXA     GFNX,4
01242 -0600 00 0 01103           STQ     FILNAM+1        THIS IS JUST THE POINTER BEING SAVED
01243  0074 00 4 01216           TSX     GETBCD,4        GET PRIMARY NAME
01244  0602 00 0 01102           SLW     FILNAM          ..
01245  0500 00 0 01103           CLA     FILNAM+1        GET SECONDARY NAME
01246  0074 00 4 01216           TSX     GETBCD,4        ..
01247  0602 00 0 01103           SLW     FILNAM+1        ..
01250  0774 00 4 00000  GFNX     AXT     **,4            RETURN TO CALLER
01251  0020 00 4 00001           TRA     1,4             ..
                         *
                         *FILESEEK ROUTINE USES  OPEN AND BUFFER IN NEW IO FILL SYSTEM
                         *
01252  0634 00 4 01256  FILSEK  SXA     FILSA,4
01253  0074 00 4 01241           TSX     GTFLNM,4        SET UP FILNAM
```

```
01254  0074 00 4 01155          TSX     $SEEK,4
01255 -0754 00 0 00000          ZAC
01256  0774 00 4 00000  FILSA   AXT     **,4
01257  0020 00 4 00001          TRA     1,4
                        *
                        *FILENDRD USES CLOSE ROUTINE OF NEW FILE SYSTEM
                        *
01260  0634 00 4 01271  FENDRD SXA     FENDA,4
01261  0074 00 4 01241          TSX     GTFLNM,4        SET UP FILNAM
01262                            CALLIO  CLOSE,FILNAM,DISKER CLOSE THE FILE
01267  0074 00 4 05452          TSX     TREAD,4
01270 -0754 00 0 00000          ZAC
01271  0774 00 4 00000  FENDA   AXT     **,4
01272  0020 00 4 00001          TRA     1,4
                        *
                        *FILEDELETE ROUTINE USES DELFIL OF NEW IO FILE SYSTEM
                        *
01273  0634 00 4 01310  FILDEL SXA     FILA,4
01274 -0600 00 0 01313          STQ     FILB            IF THIS ONE USED GTFLNM, FILNAM WOULD DIE
01275  0074 00 4 01216          TSX     $GETBCD,4
01276  0602 00 0 01312          SLW     FILC
01277  0500 00 0 01313          CLA     FILB
01300  0074 00 4 01216          TSX     $GETBCD,4
01301  0602 00 0 01313          SLW     FILB
01302                            CALLIO  DELFIL,FILC,*+1
01307 -0754 00 0 00000          ZAC
01310  0774 00 4 00000  FILA    AXT     **,4
01311  0020 00 4 00001          TRA     1,4
01312  0 00000 0 00000  FILC    PZE
01313  0 00000 0 00000  FILB    PZE

                        *       THE NOFILE ROUTINE RETURNS *T* IF THE NAMED FILE IS MISSING,
                        *  NIL OTHERWISE.

01314  0634 00 4 01332  NOFILE SXA     NOFILX,4
01315 -0600 00 0 01313          STQ     FILB            ONCE AGAIN, GTFLNM MIGHT CLOBBER READ
01316  0074 00 4 01216          TSX     GETBCD,4        ..
01317  0602 00 0 01312          SLW     FILC            ..
01320  0500 00 0 01313          CLA     FILB
01321  0074 00 4 01216          TSX     GETBCD,4        ..
01322  0602 00 0 01313          SLW     FILB            ..
01323                            CALLIO  FSTATE,FILC,(COMMON,NOFILY) ERROR MEANS NOFILY MEANS OK
01331 -0754 00 0 00000          ZAC                     *FILE WAS THERE.  RETURN NIL
01332  0774 00 4 00000  NOFILX AXT     **,4            RETURN TO CALLER
01333  0020 00 4 00001          TRA     1,4             ..
01334  0500 00 0 00126  NOFILY CLA     QD1             *RETURN TRUE, SINCE FILE GONE
01335  0020 00 0 01332          TRA     NOFILX          ..
                        *
                        *FILE1 FOR NEW FILE SYSTEM
                        *
01336 -0600 00 0 01427  FILE1   STQ     FFX
01337  0601 00 0 01426          STO     FFZ
01340  0500 00 0 02531          CLA     $ALIST+3
01341  0601 00 0 01424          STO     FC
01342  0500 00 0 02532          CLA     $ALIST+4
```

```
01343  0601 00 0 01425          STO FD
01344  0500 00 0 01426          CLA     FFZ
01345  0634 00 4 01422          SXA     FFY,4
01346  0074 00 4 01216          TSX     $GETBCD,4
01347  0602 00 0 01426          SLW     FFZ
01350  0500 00 0 01427          CLA     FFX
01351  0074 00 4 01216          TSX     $GETBCD,4
01352  0602 00 0 01427          SLW     FFX
01353 -0520 00 0 01056          NZT     $OUTSW
01354  0020 00 0 01366          TRA     FA
01355                           CALLIO  CLOSE,OUTFIL,*+1
01362 -0500 00 0 01104          CAL     $OUTFIL
01363  0602 00 0 01430          SLW     FFU
01364 -0500 00 0 01105          CAL     $OUTFIL+1
01365  0602 00 0 01431          SLW     FFV
01366  0500 00 0 01056  FA      CLA     $OUTSW
01367  0601 00 0 01432          STO     FFW
01370 -0625 00 0 01056          STL     $OUTSW
01371 -0500 00 0 01426          CAL     FFZ
01372  0602 00 0 01104          SLW     $OUTFIL
01373 -0500 00 0 01427          CAL     FFX
01374  0602 00 0 01105          SLW     $OUTFIL+1
01375  0500 00 0 01425          CLA     FD
01376  0100 00 0 01401          TZE     FAA
01377  0074 00 4 01115          TSX     $APEND,4
01400  0020 00 0 01402          TRA     FAB
01401  0074 00 4 01106  FAA     TSX     $ASIGN,4
01402  0500 00 0 01424  FAB     CLA     FC
01403  0074 00 4 04327          TSX     $PUNCH,4
01404                           CALLIO  CLOSE,OUTFIL,DISKER
01411 -0500 00 0 01430          CAL     FFU
01412  0602 00 0 01104          SLW     $OUTFIL
01413 -0500 00 0 01431          CAL     FFV
01414  0602 00 0 01105          SLW     $OUTFIL+1
01415  0500 00 0 01432          CLA     FFW          CHANGE DUE TO ERROR IN FILE1
01416  0601 00 0 01056          STO     OUTSW
01417  0100 00 0 01421          TZE     FB
01420  0074 00 4 01115          TSX     $APEND,4
01421 -0754 00 0 00000  FB      ZAC
01422  0774 00 4 00000  FFY     AXT     **,4
01423  0020 00 4 00001          TRA     1,4
01424  0 00000 0 00000  FC      PZE
01425  0 00000 0 00000  FD      PZE
01426  0 00000 0 00000  FFZ
01427  0 00000 0 00000  FFX
01430  0 00000 0 00000  FFU
01431  0 00000 0 00000  FFV
01432  0 00000 0 00000  FFW
                        *
                        *ROUTINE WHICH PRINTS BACKTRACE OF LAST ERROR
                        *
01433  0500 00 0 01435  BACKTR CLA     *+2
01434  0020 00 4 00001          TRA     1,4
01435  0 00000 0 00000          PZE
                        *
                        * USED IN CONJUNCTION WITH SETBK FOR HANDLING INTERRUPTS
```

```
                                        *
01436  0634 00 4 01441  SAVBK  SXA     SAVSY,4
01437                    TSSX    SAVBRK,4
01440 -0754 00 0 00000           ZAC
01441  0774 00 4 00000  SAVSY  AXT     **,4
01442  0020 00 4 00001          TRA     1,4
                                        *
                                        *ROUTINE WHICH HANDLES INTERRUPTS FROM THE TYPEWRITER
                                        *
01443  0634 00 4 01446  SETBK  SXA     SETX,4
01444                    TSSX    SETBRK,4
01445  0 00000 0 01451          PZE     SETB
01446  0774 00 4 00000  SETX   AXT     **,4
01447 -0754 00 0 00000           ZAC
01450  0020 00 4 00001          TRA     1,4
01451 -0520 00 0 02502  SETB   NZT     $RCSGNN
01452  0020 00 0 01470          TRA     SETE
01453  0634 00 4 01463          SXA     SETY,4
01454  0765 00 0 00001          LRS     1
01455  0602 00 0 01475          SLW     SETA
01456 -0600 00 0 01476          STQ     SETC
01457                    TSSX    GETBRK,4
01460  0621 00 0 01467          STA     SETG
01461 -0774 00 4 01467          AXC     SETE-1,4
01462  0634 00 4 02265          SXA     E$RCX,4
01463  0774 00 4 00000  SETY   AXT     0,4
01464 -0500 00 0 01475          CAL     SETA
01465  0560 00 0 01476          LDQ     SETC
01466  0763 00 0 00001          LLS     1
01467  0020 00 0 00000  SETG   TRA     **
01470                    SETE   TSSX    SETBRK,4
01471  0 00000 0 01451          PZE     SETB
01472  0074 00 4 05452          TSX     $TEREAD,4
01473  0500 00 0 00174          CLA     H40D
01474  0074 00 4 01670          TSX     $ERROR1,4
01475  0 00000 0 00000  SETA   PZE
01476  0 00000 0 00000  SETC   PZE
                                        *
                                        * THIS ROUTINE EXPLODES ...
                                        *
                                        * AN S-EXPRESSION
                                        *(TOO BAD)
01477 -0625 00 0 01522  EXPLOD STL     XPLSW
01500  0634 00 4 01505          SXA     XPLA,4
01501  0074 00 4 03564          TSX     PRINT,4
01502  0500 00 0 02673          CLA     XPLE
01503  0600 00 0 01522          STZ     XPLSW
01504  0600 00 0 02673          STZ     XPLE          ***CHANGE IN MARTIN'S LOGIC***
01505  0774 00 4 00000  XPLA   AXT     **,4
01506  0020 00 4 00001          TRA     1,4
01507  0634 00 4 01520  XPLG   SXA     XPLX,4
01510  0601 00 0 01523          STC     XPLS
01511  0774 00 4 01523          AXT     XPLS,4
01512 -0756 00 4 00000          PCD     0,4
01513  0074 00 4 11046          TSX     UNPACK,4
01514  0560 00 0 02673          LDQ     XPLE
```

```
01515  0131 00 0 00000          XCA
01516  0074 00 4 07562          TSX     NCONC,4
01517  0601 00 0 02673          STO     XPLE
01520  0774 00 4 00000  XPLX    AXT     0,4
01521  0020 00 4 00001          TRA     1,4
01522  0 00000 0 00000  XPLSw   PZE
01523  0 00000 0 00000  XPLS    PZE
                         INSERT  $ZERRRF
```

ERROR PROCESSING

```
                                *
                                * ERROR        PROCESSES ALL LISP ERRORS. NORMALLY GIVES ERROR NUMBER,
                                *              ERROR LOCATION, LISP PRINT OF AC AND BACK TRACE OFALL
                                *              FUNCTIONS ENTERED ON PUSH DOWN LIST.
                                *
01524  0 00000 0 00000   ERAC                                PLACE TO STORE MACHINE REGISTERS
01525  0 00000 0 00000   ERMQ
01526  0 00000 0 00000   ERIND
01527  0 00000 0 00000   ERX                                 INDEX 1,,INDEX 2
01530  3 00000 0 01531   ERROR  TXH     **+1,,**             INDEX 4
01531 -0520 00 0 10442          NZT     ERNULL               SEE IF ERROR PROGRAM IS TO BE EXECUTED
01532  0522 00 0 10443          XEC     EREXIT               NORMAL SETTING GOES TO EVALQUOTE
01533 -0600 00 0 01525          STQ     ERMQ                 SAVE MACHINE REGISTERS
01534  0604 00 0 01526          STI     ERIND
01535  0634 00 1 01527          SXA     ERX,1
01536 -0634 00 2 01527          SXD     ERX,2
                                * PLAY WITH SENSE INDICATOR DELETED
01537  0601 00 0 01611          STO     ERT                  AC TO BE PRINTED
01540  0500 00 4 00001          CLA     1,4                  ERROR NUMBER TO AC
01541  0601 00 0 01614          STO     ERM                  PUT IN ERROR MESSAGE
01542                           TSSX    WRFLX,4              WRITE OUT ERROR MESSAGE
01543  0 00003 0 01612                  ERO,,3
01544  0520 00 0 01610          ZET     BACACT               SKIP IF BACK TRACE IS NOT ACTIVE
01545  0020 00 0 01604          TRA     BACER                GO TO SPECIAL ROUTINE
01546 -0625 00 0 01610          STL     BACACT               MAKE BACK TRACE ROUTINE ACTIVE
01547  0500 00 0 01611          CLA     ERT                  PICK UP AC ON ENTRANCE
01550  0074 00 4 03564          TSX     $PRINT,4             PRINT IT IN LISP
01551  0054 00  000200          RFT     NOBACT               TEST FOR NO BACK TRACE
01552  0020 00 0 01602          TRA     BACD                 GO TO EXIT
01553  0560 00 0 15244          LDQ     =0                   ZERO THE ERROR LIST
01554 -0534 00 4 10446          LXD     NUBPDL,4             BEGINNING OF PUSH DOWN LIST
01555  1 77777 4 01556          TXI     *+1,4,-1             PUSH UP BY -1
01556 -0634 00 4 01560          SXD     BEX,4                SET UP ALL DONE TEST INSTRUCTION
01557 -0534 00 4 01727          LXD     $CPPI,4              PICK UP CURRENT PDL COUNTER
01560  3 00000 4 01600  BEX    TXH     BACTD,4,**           GO IF ALL UNSAVED
01561 -0500 00 4 77777          CAL     -1,4                 EITHER UNSAVE OR UNWND
01562 -0320 00 0 00134          ANA     $PMASK               DEPENDING ON COMPILED OR
01563  0322 00 0 00130          ERA     $QP5                 SYSTEM SUBROUTINE PUT IT THERE
01564  0100 00 0 01567          TZE     *+3                  TEST IS FOR STR OP
01565  0074 00 4 01736          TSX     UNSAVE,4             IN LAST WORD OF BLOCK FROM COMPILER
01566  0020 00 0 01570          TRA     *+2
01567  0074 00 4 15027          TSX     C$UNWND,4
01570 -0534 00 4 01727          LXD     $CPPI,4              BEGINNING OF BLOCK JUST UNSAVED
01571  0500 00 4 00000          CLA     0,4                  LAST IR 4 WORD
01572  0734 00 4 00000          PAX     0,4                  FUNCTION ATOMIC SYMBOL
01573 -0734 00 4 00000          PXD     0,4                  PUT IN DECREMENT
01574  0074 00 4 02714          TSX     $CONS,4              ADD TO ERROR LIST
01575  0131 00 0 00000          XCA                          ANSWER TO MQ
01576 -0534 00 4 01727          LXD     $CPPI,4              PUSH DOWN INDICTOR
01577  0020 00 0 01560          TRA     BEX                  GO BACK FOR NEXT
01600  0131 00 0 00000  BACTD  XCA                          LIST TO AC
01601  0601 00 0 01435          STO     BACKTR+2       WILL NOT PRINT THE TRACE UNLESS DESIRED
01602  0600 00 0 01610  BACD   STZ     BACACT               DE-ACTIVATE THE BACK TRACE ROUTINE
01603  0522 00 0 10443          XEC     EREXIT               NORMAL SETTING GOES TO EVALQUOTE
                                *
```

ERROR PROCESSING

```
01604                          BACER  TSSX    WRFLX,4        *BACK TRACE CAUSED ERROR OF ITS OWN
01605   0 00007 0 01615                       BACE,,7
01606   0074 00 4 04230                TSX     TERPRI,4       ***THIS SHOULD CLEAR PRINT BUFFER
01607   0020 00 0 01602                TRA     BACD                RESET AND RETURN
                     000200  NOBACT BOOL    200                 NO BACK TRACE INDICATOR
01610   0 00000 0 00000        BACACT                             NON-ZERO MEANS BACK TRACE ACTIVE
01611   0 00000 0 00000         ERT                               TEMPORARY STORAGE FOR AC
01612   545454606025           ERO     BCI     2,***  ERROR
01614   0 00000 0 00000         ERM                               ERROR NUMBER IN BCD GOES HERE
01615   545454602122            BACE   BCI     7,*** ABOVE ERROR TERMINATED BACK-TRACE ***

                               *       THE USER CAN ALWAYS LAP IN HIS OWN DCT-HANDLER.  I CAN'T SEE WHY
                               *  HE'D WANT TO, THOUGH.

01624   0020 00 4 00001        DCT     TRA     1,4            IGNORE DCTS, I SAY

                               *       FLOATING-POINT TRAPS ARE ANOTHER MATTER.  HERE IS AN OLD SHARE
                               *  FPT.

01625   0604 00 0 01666        FLAPTR STI      SI
01626   0601 00 0 01667                STO     FLPTMP         P AND Q ARE LOST ANYWAY
01627   0500 00 0 00000                CLA     00000          WE NEED RETURN ADDRESS, WHICH WON'T
01630   0621 00 0 01650                STA     FLPOUT          BE THERE WHEN WE RETURN
01631   0044 00 0 00000                PAI                    NOW WE WILL LOOK AT TRAP-TYPE BITS
01632   0500 00 0 01667                CLA     FLPTMP         WELL, WE ALMOST HAVE THE AC BACK
01633  -0056 00  000002                LNT     2              WAS PROBLEM IN THE AC
01634   0020 00 0 01654                TRA     FLAPMQ         *NO
01635   0760 00 0 00000                CLM                    ASSUME UNDERFLOW
01636  -0054 00  000004                LFT     4              WAS IT REALLY OVERFLOW
01637   0760 00 0 00006                COM                    *YES
01640   0771 00 0 00002                ARS     2              CLEAR P, Q
01641  -0054 00  000010                LFT     10             WAS IT SOMETHING TO DO WITH DIVISION
01642   0020 00 0 01651                TRA     FLAPDV         *YES
01643   0601 00 0 01667  FLAPEX STO     FLPTMP         *NO, AC IS OK NOW
01644   0560 00 0 01667                LDQ     FLPTMP         SET MQ SIMILARLY (THIS IS A PRETTY CRUDE
01645   0441 00 0 00076  FLPRTN LDI     FLAPCZ          ROUTINE, ISN'T IT)
01646   0604 00 0 00000                STI     00000          THIS IS THE NORMAL CONTENTS OF 00000
01647   0441 00 0 01666                LDI     SI             RESTORE MACHINE CONDITIONS
01650   0020 00 0 00000  FLPOUT TRA     **             RETURN TO USER

01651  -0056 00  000001  FLAPDV LNT     1              WAS AC DIVIDE.  HOW IS MQ
01652   0020 00 0 01645                TRA     FLPRTN         *OK, SO RETURN TO USER
01653   0020 00 0 01643                TRA     FLAPEX         *NG, USE AC VALUE

01654   0131 00 0 00000  FLAPMQ XCA             AC OK, BUT MQ IS BAD
01655   0760 00 0 00000                CLM                    PLAY THE SAME GAME AS ABOVE WITH THE AC
01656  -0056 00  000004                LNT     4              WAS IT OVERFLOW
01657   0020 00 0 01662                TRA     FLPMQ1         *NO, GO TO CHECK FOR DIVIDE
01660   0760 00 0 00006                COM                    *YES, LEAVE BIG RESULT
01661   0020 00 0 01645                TRA     FLPRTN         RETURN TO USER
01662  -0054 00  000010  FLPMQ1 LFT     10             WAS MQ UF OR DIVIDE
01663   0020 00 0 01643                TRA     FLAPEX         *DIVIDE, USE 0 VALUE
01664   0131 00 0 00000                XCA                    *UF, USE DIFFERENT 0
01665   0020 00 0 01645                TRA     FLPRTN
```

ERROR PROCESSING

```
01666   0 00000 0 00000  SI
01667   0 00000 0 00000  FLPTMP

                         ERROR1 USED BY APPLY HAS ONE ARGUMENT AND PRINTS IT USING
                         PRINT.

01670                    ERROR1 ERROR   CALLED        APPLIED FUNCTION CALLED ERROR
```

ERROR PROCESSING

```
                                    INSERT  $SETCNF
                                    HEAD    0
                                *
                                *           STORAGE MAP CELLS FOR LISP
                                *
01673   0 00000 0 70240     XORG            UPERML              ORIGIN OF EXCISED BINARY PROGRAM
01674   0 00000 0 70237      TFS            UPERML-1            UPPER LIMIT OF FREE STORAGE
01675   0 00000 0 64260      MFS            LOWERP              LOW LIMIT OF PERM. LIST STRUCTURE
01676   0 00000 0 00000      BFS                                BOTTOM OF FREE STORAGE
01677   0 00000 0 00000      TBT                                TOP OF BIT TABLE
01700   0 00000 0 00000      BBT                                BOTTOM OF BIT TABLR
01701   0 00000 0 00000      TFW                                TOP OF FULL WORD SPACE PROPER
01702   0 00000 0 00000      BFW                                BOTTOM OF FULL WORD SPACE
01703   0 00000 0 00000      TPG            **
01704   0 00000 0 00000      ORG                                ORIGIN OF BINARY PROGRAM IN DECREMENT
01705 +000000014000      LBINPG OCT        14000          ***LENGTH OF BINARY PROG SPACE***
01706 +000000005000      LPBPDL OCT        5000           ***LENGTH OF PDL****
01707 +000000004220      LFULWS OCT        4220           ****LENGTH OF FWS+BIT TABLE***
01710 +000000024000      LFREES OCT        24000          ***LENGTH OF FREE STORAGE***
01711   0 00000 0 00000      LBT                                LENGTH OF FULL WORD BIT TABLE
                                *
                                *       CELLS TO DETERMINE WHERE PROGRAMS AND ARRAYS GO
                                *
01712   0 00000 0 00000     BPLACE          0                   NOISE MEANS ASSEMBLE TO EXCISABLE AREA
01713   0 00000 0 00000     EXCISD          0                   NOISE MEANS EXCISION HAS MEAD ABOVE IM
                                *                               POSSIBLE
                                *
                                *       EXCISABLE (T)   SETS LAP AND ARRAYS TO GO INTO EXCISABLE SPACE
                                *       EXCISABLE(F) SETS THEM TO GO INTO BPS AS IS NORMAL
                                *
                                    HEAD    X
01714  0520 00 0 01713   EXCABL ZET     EXCISD              CAN WE STILL PUT THINGS UP THERE
01715 -0100 00 0 01720         TNZ     EXER                NO, BUT WE WERE TRYING
01716  0601 00 0 01712         STO     BPLACE              SET FLAG
01717  0020 00 4 00001         TRA     1,4
01720                  EXER    ERROR   XBLGON          EXCISABLE SPACE HAS BEEN EXCISED

                                    HEAD    0
                                * SAVE AND UNSAVE                   THE CLOSED SUBROUTINES THAT CONTROL
                                *                   THE PUBLIC PUSH DOWN LIST. THE CALLING SEQUENCES ARE ...
                                *
                                *       TSX     $SAVE,4
                                *       TXL     $ENDN,,END OF BLOCK TO BE SAVED  + 2
                                *       RETURN
                                *                   WHERE N IN $ENDN IS THE NUMBER OF ITEMS TO BE SAVED
                                *
                                *       TSX     UNSAVE,4
                                *       RETURN
                                *                   THE SAVED ITEMS MUST BE IN A CONTIGOUS BLOCK WITH THE
                                *                   THE FIRST ITEM    PZE ATOMIC NAME OF SUBR,,IR 4
                                *                   THE SAVE PARAMETER WORD IS ADDED AS THE LAST ITEM ON THE
                                *                   BLOCK TO BE SAVED BUT IS NOT UNSAVED.
                                *
01723  0634 00 2 01774    SAVE  SXA     SAVY,2              SAVE INDEX 2 AND 1
01724  0601 00 0 01776         STO     SAVT                SAVE THE AC
```

ERROR PROCESSING

```
01725  0500 60 4 00001          CLA*   1,4        AMMOUNT TO SUBTRACT FROM CPPI IN AC
01726  0734 00 1 00000          PAX    0,1        PUT - NUMBER OF ITEMS TO BE SAVED + 1
01727  1 00000 1 01730   CPPI   TXI    *+1,1,**   IN IR 1 AND INCREMENT BE PUSH DOWN CNT
01730 -3 00000 1 02004          TXL    NOPDL,1,** GO TO NOPDL IF NOT ENOUGH PDL
01731 -0634 00 1 01727          SXD    $CPPI,1    UP DATE PDL COUNTER LOCATION
01732  0500 00 4 00001          CLA    1,4        PARAMETER WORD
01733  0601 00 1 77777          STO    -1,1       PUT ON PUSH DOWN LIST
01734 -0737 00 2 00000          PDC    0,2        LOCATION OF BLOCK TO BE SAVED + 2
01735  0522 00 4 00001          XEC    1,4        JUMP INTO SAVE TABLE
                        *
01736  0634 00 2 01774  UNSAVE SXA    SAVY,2     SAVE INDEX 2 AND 1
01737  0601 00 0 01776          STO    SAVT       SAVE THE AC
01740 -0534 00 2 01727          LXD    $CPPI,2    CURRENT PUSH DOWN COUNTER
01741  0500 00 2 77777          CLA    -1,2       LAST SAVE PARAMETER WORD
01742  0621 00 0 01745          STA    SAVJ       SET FETCH AND TXI INSTRUCTIONS
01743  0621 00 0 01751          STA    SAVK
01744 -0634 00 2 01746          SXD    SAVI,2     SET UP TO RESTORE PDL COUNTER
01745  0535 00 1 00000   SAVJ   LAC    **,1       NUMBER TO BE UNSAVED
01746  1 00000 1 01747   SAVI   TXI    *+1,1,**   ADD PUSH DOWN COUNTER
01747 -0634 00 1 01727          SXD    $CPPI,1    UPDATE PDL COUNTER CELL
01750 -0737 00 1 00000          PDC    0,1        LOCATION OF END OF BLOCK + 2
01751  1 00001 4 00000   SAVK   TXI    **,4,1     JUMP TO PUSH DOWN TABLE AND SET IR 4
                        *                         PROPER EXIT .
                        *
                        *                  SAVE AND UNSAVE TABLE TO DO THE ACTUAL MOVING TO AND FROM
                        *                  THE PUBLIC PUSH DOWN LIST.
                        *
01752  0500 00 2 77767   END8   CLA    -9,2
01753  0601 00 1 77767          STO    -9,1
01754  0500 00 2 77770   END7   CLA    -8,2
01755  0601 00 1 77770          STO    -8,1
01756  0500 00 2 77771   END6   CLA    -7,2
01757  0601 00 1 77771          STO    -7,1
01760  0500 00 2 77772   END5   CLA    -6,2
01761  0601 00 1 77772          STO    -6,1
01762  0500 00 2 77773   END4   CLA    -5,2
01763  0601 00 1 77773          STO    -5,1
01764  0500 00 2 77774   END3   CLA    -4,2
01765  0601 00 1 77774          STO    -4,1
01766  0500 00 2 77775   END2   CLA    -3,2
01767  0601 00 1 77775          STO    -3,1
01770  0500 00 2 77776   END1   CLA    -2,2
01771  0601 00 1 77776          STO    -2,1
01772  0500 00 0 01776   END0   CLA    SAVT       RESTORE THE AC
01773 -0534 00 1 01727          LXD    $CPPI,1    MAKE IX1 AND $CPPI AGREE
01774  0774 00 2 00000   SAVY   AXT    **,2
01775  0020 00 4 00002          TRA    2,4        EXIT
                        *
01776  0 00000 0 00000   SAVT                     TEMPORARY STORAGE FOR AC
                        *         TIMING INFORAMTION .. SAVE AND UNSAVE 34 + 4N CYCLES
                        *         ON THE 709  (SUBTRACT 5 CYCLES FOR SAVE AND 4 FOR UNSAVE
                        *         ON THE 7090)
                        *
```

ERROR PROCESSING

TERPDL
RESETS PUBLIC PUSH DOWN LIST TO ZERO

```
01777   0500 00 0 02002   TERPDL CLA  $CSSI
02000   0622 00 0 01727          STD  CPPI
02001   0020 00 4 00001          TRA  1,4
02002   0 00000 0 00000    CSSI
02003  -3 00000 4 02004   ENDPDL TXL      **+1,4,**          OUT OF PDL TEST INSTRUCTION (IS XEC)
02004                      NOPDL  ERROR    PDLOUT         OUT OF PUBLIC PUSH DOWN LIST
```

ERROR PROCESSING

```
                                             INSERT   $GCOLEF
                                        *
                                             HEAD     E              NECESSARY IF CNSFWL IS GONE****
                                        *
                                        *
                                        * RECLAIMER   LISP 1.5 STORAGE CONTROL PROGRAM.  CODED 1 MARCH 1961
                                        *
02007    0634 00 4 02265  RECLAM SXA    RCX,4                  SAVE INDEX REGISTERS
02010    0634 00 2 02266         SXA    RCY,2
02011    0634 00 1 02267         SXA    RCZ,1
02012    0604 00 0 02434         STI    RCIND                  AND MACHINE REGISTETS
02013   -0625 00 0 02502         STL    RCSGNN
02014    0601 00 0 02516         STO    RCAC
02015   -0600 00 0 02517         STQ    RCMQ
02016    0600 00 0 02435         STZ    RCBE                   INITIALIZE BAD EXIT CELL
02017    0560 00 0 02500         LDQ    RCSGNL                 SIGNAL PHASE 1
02020    0774 00 4 00000  A      AXT    **,4                   BIT TABLE LENGTH
02021    0600 00 4 00000  B      STZ    **,4                   BOTTOM FREE STORAGE
02022    2 00001 4 02021         TIX    *-1,4,1                ZERO THE BIT TABLE
02023   -0534 00 4 02003         LXD    ENDPDL,4               END OF PDL
02024   -0634 00 4 02026         SXD    RCIA,4                 SET UP TNX INSTRUCTION
02025   -0534 00 4 01727         LXD    $CPPI,4                CURRENT PUSH DOWN LIST LOC.
02026   -2 00000 4 02325  RCIA   TNX    MLPDE,4,**             AMMOUNT OF PUSH DOWN LIST AVAILABLE
02027   -0634 00 4 02324         SXD    MLPDC,4                SET CELL IN MRKLST
02030    0634 00 4 02203         SXA    ZPDLA,4                LENGTH LEFT BAR FOR ZEROING PDL
02031   -0774 00 2 64457         AXC    OBLIST,2               POINTER TO OBJECT LICT
02032    0074 00 4 02310         TSX    MRKLST,4               MARK THE LIST
                                        *
                                        *              TEMLIS MARKER
                                        *              TEMLIS IS A LIST IN FREE STORAGE AND FULL WORD SPACE
                                        *              OF THE FORM (CONS (CONSW BEG,,END) TEMLIS) AND INDICATES
                                        *              PLACES WHERE LIST STRUCTURE MAY BE DURING A GARBAGE
                                        *              COLLECTION. USED PRINCIPALLY BY THE COMPILER
                                        *
02033    0600 00 0 02503         STZ    TMLM                   SET EXIT SWITCH
02034   -0534 00 4 02514         LXD    TEMLIS,4
02035    0500 00 4 00000  TMLJ   CLA    0,4                    NEXT WORD ON TEMLIS
02036    0622 00 0 02503         STD    TMLM                   SAVE POINTER TO NEXT WORD
02037    0734 00 4 00000         PAX    0,4                    POINTER TO FULL WORD
02040    0500 00 4 00000         CLA    0,4                    FULL WORD
02041    0734 00 4 00000         PAX    0,4                    BEGINNING OF ARRAY
02042   -0634 00 4 02046         SXD    TMLD,4
02043   -0734 00 1 00000         PDX    0,1                    END OF ARRAY
02044    1 00001 1 02045         TXI    *+1,1,1                ADD 1
02045    0634 00 1 02050  TMLK   SXA    TMLE,1
02046   -2 00000 1 02062  TMLD   TNX    TMLH,1,**              SUBTRACT BEGINNING , GIVES COUNT IN IR
02047    0634 00 1 02424         SXA    GCPDLC,1               LAST USE IS MARKING PDL, SAVE LENGTH
02050    0441 00 1 00000  TMLE   LDI    **,1                   PICK UP WORD
02051    0444 00 0 02504         OFT    TMPTM                  SKIP IF NOTAG OR PREFIX
02052    0020 00 0 02061         TRA    TMLG                   NOT A LIST, DO NOT MARK
02053   -0046 00 0 00000         PIA                           ITEM TO AC
02054    0621 00 0 02057         STA    TMLF                   SAVE ADDRESS
02055   -0734 00 2 00000         PDX    0,2
02056    0074 00 4 02310         TSX    MRKLST,4               MARK THE DECREMENT
02057    0774 00 2 00000  TMLF   AXT    **,2                   ADDRESS OF WORD TO IR
```

49322

```
02060  0074 00 4 02310          TSX   MRKLST,4          MARK IT
02061  2 00001 1 02050  TMLG    TIX   TMLE,1,1          GET NEXT WORD IN ARRAY
02062 -0534 00 4 02503  TMLH    LXD   TMLM,4           NEXT TEMLIS ITEM
02063  3 00000 4 02035          TXH   TMLJ,4,0          GO IF NOT DONE
02064  0520 00 0 02503          ZET   TMLM              TEST FOR EXIT
02065  0020 00 0 02074          TRA   MPDLF             ALL DONE
02066 -0535 00 4 02002          LDC   $CSSI,4           BEGINNING OF PDL
02067 -0634 00 4 02046          SXD   TMLD,4            SET UP CELL
02070 -0535 00 1 01727          LDC   $CPPI,1           FIRST FREE CELL ON PDL
02071 -0625 00 0 02503          STL   TMLM              INDICATE LAST USE OF LOOP
02072  0600 00 0 02424          STZ   GCPDLC            PUSH DOWN LENGTH INITIALLY ZERO
02073  0020 00 0 02045          TRA   TMLK              GO MARK PUSH DOWN LIST
02074 -0534 00 4 02515  MPDLF   LXD   ARYLIS,4         START TO MARK ACTIVE ARRAYS
02075 -3 00000 4 02132  MARYB   TXL   RCB,4,0          GO IF NO ARRAYS
02076  0500 00 4 00000          CLA   0,4               NEXT WORD ON ARYLIS
02077  0622 00 0 02436          STD   MARYT             SAVE POINTER TO NEXT WORD
02100  0734 00 4 00000          PAX   0,4               ARYATOM TO AC
02101  0500 00 4 00000  MARYA   CLA   0,4               NEXT WORD ON ATOM
02102  0734 00 4 00000          PAX   0,4
02103 -3 13222 4 02105          TXL   *+2,4,$ARRAY-1    SERCH FOR ARRAY SPECIFICATION
02104 -3 13223 4 02111          TXL   MRKA,4,$ARRAY     GO IF FOUND
02105 -0734 00 4 00000          PDX   0,4               POINTER TO NEXT WORD
02106  3 00000 4 02101          TXH   MARYA,4,0         GO IF NOT END OF ATOM
02107 -0534 00 4 02436  MARYC   LXD   MARYT,4           NEXT WORD ON ARYLIS
02110  0020 00 0 02075          TRA   MARYB
                        *
02111 -0734 00 4 00000  MRKA    PDX   0,4               GET ARRAY SPECIFICATIONS
02112  0500 00 4 00000          CLA   0,4
02113  0734 00 4 00000          PAX   0,4
02114  0500 00 4 00000          CLA   0,4
02115  0734 00 4 00000          PAX   0,4
02116  0500 00 4 00000          CLA   0,4               FIRST SPEC. WORD
02117  0734 00 2 00000          PAX   0,2               END OF ARRAY + 1
02120  0621 00 0 02125          STA   MRKE              END OF ARRAY + 1
02121  0500 00 4 00001          CLA   1,4
02122  0601 00 0 02505          STO   MRKP              SECOND SPEC. WORD  TOTAL L,, LIST L
02123 -0534 00 1 02505          LXD   MRKP,1            GET LIST LENGTH IF ANY
02124 -3 00000 1 02107          TXL   MARYC,1,0         EXIT IF A NON-LIST ARRAY
02125  0500 00 1 00000  MRKE    CLA   **,1              LIST ITEM
02126 -0734 00 2 00000          PDX   0,2
02127  0074 00 4 02310          TSX   MRKLST,4          MARK IT
02130  2 00001 1 02125          TIX   MRKE,1,1          GET NEXT ITEM
02131  0020 00 0 02107          TRA   MARYC             EXIT
                        *
                        *               ALL MARKING DONE. NOW SWEEP FREE STORAGE
                        *
02132  0774 00 2 00000  RCB     AXT   0,2               ZERO COUNT IR
02133  0600 00 0 02426          STZ   FSC               INITIALIZE COUNTER
02134  0560 00 0 02501          LDQ   RCSGNM            SWEEPING SIGNAL TO MQ
02135 -0774 00 1 02735          AXC   $FREE,1           INITIALIZE LAST LOC IR
02136 -0774 00 4 00000  F       AXC   **,4              TOP FREE STORAGE
02137  0502 00 4 00000  SFSL    CLS   0,4               PICK UP WORD
02140 -0120 00 0 02147          TMI   SFSC              COLLECT IF SIGN NOW MINUS
02141  0601 00 4 00000          STO   0,4               RESTORE WORD WITH + SIGN
02142  1 00001 4 02143  SFSA    TXI   *+1,4,1           INCREMENT BY ONE
```

```
02143 -3 00000 4 02137   G       TXL     SFSL,4,**       LOOP IF LESS THAN BOTTOM FREE STORAGE
02144  0600 00 1 00000           STZ     0,1             ZERO LAST WORD COLLECTED
02145  0634 00 2 02426           SXA     FSC,2           SAVE COUNT
02146  0020 00 0 02153           TRA     SWPFWS
02147 -0754 00 4 00000   SFSC    PXD     0,4             THIS LOCATION
02150  0601 00 1 00000           STO     0,1             STORE POINTER IN LAST WORD COLLECTED
02151 -0734 00 1 00000           PDX     0,1             UP DATE LAST WORD IR
02152  1 00001 2 02142           TXI     SFSA,2,1        UPDATE COUNTER
                         *
                         *               NOW SWEEP FULL WORD SPACE WITH THE BIT TABLE
                         *
02153  0774 00 4 02713   SWPFWS  AXT     FWORDL,4        BEGINNING OF FULL WORD LIST
02154  0634 00 4 02301           SXA     SFWA,4          INITIALIZE ADDRESS
02155  0600 00 0 02425           STZ     FWC             ZERO FULL WORD COUNTER
02156 -0774 00 1 00000   H       AXC     **,1            BOTTOM FULL WORD SPACE
02157  1 00000 1 02160   I       TXI     *+1,1,**        TOP FULL WORD SPACE
02160  0754 00 1 00000           PXA     0,1             GET ADDRESS OF BIT TABLE CORRESPONDING
02161 -0765 00 0 00005           LGR     5               TO THE BOTTOM OF FULL WORD SPACE
02162  0734 00 4 00000           PAX     0,4             BIT TABLE WORD
02163  1 00001 4 02164           TXI     *+1,4,1         MAKE INDEXING EASY
02164 -0754 00 0 00000           PXD     0,0             ZERO AC
02165 -0763 00 0 00005           LGL     5               BIT NUMBER
02166  0734 00 2 00000           PAX     0,2             INTO IR 2
02167  1 00001 2 02170           TXI     *+1,2,1         MAKE INDEXING EASY
02170  0535 00 1 02156           LAC     H,1             SET UP IR 1
02171  0441 00 4 00000   SFWLD   LDI     **,4            BOTTOM FREE STORAGE, (TBT + 1)
02172  0446 00 0 00140           ONT     MONES           SKIP IF ALL WORDS TO BE SAVED
02173  0020 00 0 02271           TRA     SFWSC           SEARCH FOR THE WORDS TO BE COLLECTED
02174  1 77740 1 02175           TXI     *+1,1,-32       DECREMENT CURRENT LOC IR
02175  2 00001 4 02171   SFWB    TIX     SFWLD,4,1       INDEX THROUGH BIT TABLE
02176  0500 00 0 02425           CLA     FWC             ALL DONE, GET FULL WORD COUNTER
02177  0601 60 0 02301           STO*    SFWA            SET UP LAST CELL COLLECTED
02200  0560 00 0 02502           LDQ     RCSGNN          PASE 3 SIGNAL
02201  0520 00 0 02430           ZET     RCT             TEST FOR OUT OF ARRAY SPACE ENTRANCE
02202  0074 00 4 02403           TSX     RELOC,4         RELOCATE AND COMPACT FULL WORD SPACE
02203  0774 00 4 00000   ZPDLA   AXT     **,4            ZERO UNUSED PDL
02204  0600 00 4 00000   ZPDL    STZ     **,4            ZERO PDL WORD
02205  2 00001 4 02204           TIX     *-1,4,1
02206  0560 00 0 00074           LDQ     CRITWN          CRITACL WORD NUMBER
02207  0600 00 0 02435           STZ     RCBE            INITIALIZE BAD EXIT TEST CELL
02210  0500 00 0 02425           CLA     FWC             NUMBER OF FULL WORDS COLLECTED
02211  0040 00 0 02213           TLQ     RCEA            TRANSFER IF MORE THAN CRITACL COLLECT
02212 -0625 00 0 02435           STL     RCBE            NOT ENOUGH, SIGNAL BAD EXIT
02213  0400 00 0 02432   RCEA    ADD     TFWC            ADD TOTAL OF FULL WORDS COLLECTED
02214  0601 00 0 02432           STO     TFWC            UPDATE COUNTER
02215 -0763 00 0 00004           LGL     4               INCREASE TOLERENCE BY 2 TO THE 4 TH
02216  0500 00 0 02426           CLA     FSC             NUMBER OF FREE STORAGE CELLS PICKED UP
02217  0040 00 0 02221           TLQ     RCEB            TRA IF GREATER THAN CRITACL NUMBER
02220 -0625 00 0 02435           STL     RCBE            NO, SIGNAL BAD EXIT
02221  0400 00 0 02433   RCEB    ADD     TFSC            ADD TOTAL OF FREE COLLECTED TO DATE
02222  0601 00 0 02433           STO     TFSC            UPDATE TOTAL
02223  0500 00 0 02427           CLA     RCC             NUMBER OF RECLAIMATION CYCLES EXECUTED
02224  0400 00 0 15245           ADD     =1              INCREMENT BY 1
02225  0601 00 0 02427           STO     RCC             UPDATE TOTAL
02226  0500 00 0 02431           CLA     RLC             NUMBER OF TIMES RELOCATION OF FWS
```

ERROR PROCESSING

```
02227  0520 00 0 02430          ZET     RCT       SKIP IF NO RELOCATION
02230  0400 00 0 15245          ADD     =1
02231  0601 00 0 02431          STO     RLC       UPDATE COUNTER
02232  0520 00 0 02435          ZET     RCBE      SKIP IF OK EXIT
02233  0020 00 0 02236          TRA     RCEC      DO VERBOSE AND BAD EXIT
02234 -0520 00 0 02477          NZT     VERBOS.   SKIP IF TALKATIVE
02235  0020 00 0 02261          TRA     RCEXIT    DO EXIT
02236  0535 00 4 02265  RCEC    LAC     RCX,4     GET EXIT IR4
02237 -0754 00 4 00000          PXD     0,4       AND CONVERT FOR PRINTING
02240  0131 00 0 00000          XCA
02241  0074 00 4 10033          TSX     OCTALP,4
02242 -0501 00 0 15304          ORA     =H 00000
02243  0602 00 0 02413          SLW     RCT1
02244  0500 00 0 02425          CLA     FWC       FULL WORD COUNTER
02245  0074 00 4 03076          TSX     $DECON,4  CONVERT TO BCD DECIMAL
02246  0602 00 0 02416          SLW     RCT4      PUT IN MESSAGE
02247  0500 00 0 02426          CLA     FSC       FREE STORAGE COUNTER
02250  0074 00 4 03076          TSX     $DECON,4  TO DECIMAL
02251  0602 00 0 02420          SLW     RCT5      PUT IN MESSAGE
02252  0500 00 0 02424          CLA     GCPDLC    NUMBER OF ACTIVE REGISTERS ON PDL
02253  0074 00 4 03076          TSX     $DECON,4  TO DECIMAL
02254  0602 00 0 02424          SLW     RCT6      IN MESSAGE
02255                            TSSX    WRFLX,4   WRITE OUT MESSAGE
02256  0 00013 0 02412                  RCTM,,11
02257  0520 00 0 02435          ZET     RCBE      SKIP IF GOOD EXIT
02260  0020 00 0 02367          TRA     RCBEX     DO BAD EXIT
02261  0500 00 0 02516  RCEXIT CLA     RCAC      RESTORE MACHINE REGISTERS
02262  0560 00 0 02517          LDQ     RCMQ
02263  0441 00 0 02434          LDI     RCIND
02264  0600 00 0 02502          STZ     RCSGNN
02265  0774 00 4 00000  RCX     AXT     **,4      AND INDEX REGISTERS
02266  0774 00 2 00000  RCY     AXT     **,2
02267  0774 00 1 00000  RCZ     AXT     **,1
02270  0020 00 4 00001          TRA     1,4       EXIT
02271  0446 00 2 02477  SFWSC   ONT     MBIT,2    CHECK FOR CURRENT BIT
02272  0020 00 0 02277          TRA     SFWC      IS OFF, COLLECT WORD
02273  1 77777 1 02274          TXI     *+1,1,-1  IS ON, DECREMENT CURRENT LOC IR
02274  2 00001 2 02271  SFWD    TIX     SFWSC,2,1 INDEX THROUGH THE BITS
02275  0774 00 2 00040          AXT     32,2      SET UP IR WITH NUMBER OF BITS PER WORD
02276  0020 00 0 02175          TRA     SFWB      EXAMINE NEXT WORD IN BIT TABLE

02277 -0754 00 1 00000  SFWC    PXD     0,1       COLLECT THIS WORD, POINTER TO THIS WOR
02300  0400 00 0 02425          ADD     FWC       D PLUS NUMBER OF WORDS COLLECTED IN AC
02301  0601 00 0 00000  SFWA    STO     **        SET LAST WORD COLLECTED
02302  0400 00 0 15245          ADD     =1        INCREMENT NUMBER OF FULL WORDS COLLECT
02303  0621 00 0 02425          STA     FWC       SAVE FULL WORD COUNTER
02304 -0737 00 1 00000          PDC     0,1       COMPLEMENT CURRENT LOCATION
02305  0634 00 1 02301          SXA     SFWA,1    TO FORM TRUE ADDRESS FOR UPDATE STORE
02306 -0734 00 1 00000          PDX     0,1       CURRENT LOCATION POINTER
02307  1 77777 1 02274          TXI     SFWD,1,-1 DECREMENT CURRENT LOCATION AND RETURN

                         * MRKLST        THE RECURSIVE SUBROUTINE THAT DOES ALL LIST MARKING
                         *
02310  3 00000 2 02354  MRKLST TXH     MLEXT,2,**  BFW BAR, REJECT POINTERS TO PROGRAM
02311 -3 00000 2 02354          TXL     MLEXT,2,**  TFS BAR - 1, REJECT POINTERS TO LOADER
```

```
02312  0634 00 1 02352         SXA    MSRTN,1           SAVE IR 1
02313  0634 00 4 02353         SXA    MRKX,4            SAVE LINK IR
02314  0774 00 1 00001         AXT    1,1               PRESET TO ONE FOR FAST PUSH DOWN ACESS
02315  0020 00 0 02333         TRA    MLIST             DO ACTUAL MARKING
                        *
02316  0502 00 2 00000  MWIN   CLS    0,2               MARK THIS WORD IN FREE STORAGE
02317  0120 00 0 02351         TPL    MOUT              TRANSFER OUT IF ALREADY MARKED
02320  0601 00 2 00000         STO    0,2               CAR OF LIST
02321  0734 00 2 00000         PAX    0,2               CAR TO IR 2
02322  0622 00 1 00000  MLEPD  STD    **,1              ENDPDL + 1, SAVE CDR OF LIST ON PDL
02323  1 00001 1 02324         TXI    *+1,1,1           INCREMENT PUSH DOWN COUNTER
02324 -3 00000 1 02333  MLPDC  TXL    MLIST,1,**        ENDPDL - C($CPPI) BAR, GO IF NOT NOPDL
02325  0074 00 4 02355  MLPDE  TSX    RCERR,4           OUT OF PUSH DOWN LIST, FATAL ERROR
02326  454660472443            BCI    3,NO PDL -MRKLST-

02331  0500 00 1 00000  MLEPE  CLA    **,1              ENDPDL + 1, GET CDR OF LIST
02332 -0734 00 2 00000         PDX    0,2               PUT IN  IR 2
02333 -3 00000 2 02351  MLIST  TXL    MOUT,2,**         TFS BAR - 1, OUT IF NOT IN LISP STORAG
02334 -3 00000 2 02316  MLBFA  TXL    MWIN,2,**         BOTTOM FREE STORAGE BAR, IN FREE
02335 -3 00000 2 02351  MLBBJ  TXL    MOUT,2,**         BBT BAR  OUT IF POINTER TO BIT TABLE
02336 -3 00000 2 02340  MLBDW  TXL    MONE,2,**         BOTTOM FULL WORD BAR, IN FULL WORD
02337  0020 00 0 02351         TRA    MOUT              EXIT , NOT ANY OF ABOVE
                        *
02340  1 00000 2 02341  MONE   TXI    *+1,2,**          TOP FULL WORD
02341  0754 00 2 00000         PXA    0,2               CALCULATE BIT TABLE WORD AND BIT
02342 -0765 00 0 00005         LGR    5
02343  0734 00 2 00000         PAX    0,2               BIT TABLE WORD
02344 -0754 00 0 00000         PXD    0,0
02345 -0763 00 0 00005         LGL    5                 BIT TABLE BIT
02346  0734 00 4 00000         PAX    0,4
02347 -0500 00 4 02476         CAL    BIT,4             PICK UP BIT
02350 -0602 00 2 00000  MLTBT  ORS    **,2              TOP BIT TABLE, PUT IN BIT
02351  2 00001 1 02331  MOUT   TIX    MLEPE,1,1         GO BACK IF IN RECURSION
02352  0774 00 1 00000  MSRTN  AXT    **,1              OTHERWISE RESTORE IR 1
02353  0774 00 4 00000  MRKX   AXT    **,4              AND LINK IR
02354  0020 00 4 00001  MLEXT  TRA    1,4               AND EXIT
                        *
                        * RCERR         RECLAIMER FATAL ERROR DUMP ROUTINE
                        *
02355 -0634 00 4 01530  RCERR  SXD    $ERROR,4          SAVE IR 4
02356  0634 00 4 02357         SXA    *+1,4             COMPLEMENT IR 4 TO GET ERROR MESSAGE
02357 -0774 00 4 00000         AXC    **,4
02360  1 00001 4 02361         TXI    *+1,4,1           LOCATION OF ERROR MESSAGE
02361  0634 00 4 02363         SXA    RCFEM,4           BUILD OUTPUT CALL
02362                          TSSX   WRFLX,4           WRITE ERROR MESSAGE
02363  0 00003 0 00000  RCFEM         **,,3             WRITE OUT 3 WORDS
02364  0600 00 0 02735         STZ    $FREE
02365  0600 00 0 02713         STZ    FWORDL            ZERO STORAGE LISTS
                        * CHANGE MADE RIGHT HERE********
02366  0020 00 0 10145         TRA    S$EVQDN
                        *END OF CHANGE****
                        *
02367  0441 00 0 02434  RCBEX  LDI    RCIND             RESTORE MACHINE REGISTERS
02370  0600 00 0 02502         STZ    RCSGNN
02371  0500 00 0 02516         CLA    RCAC
```

```
02372  0560 00 0 02517          LDQ     RCMQ
02373  0534 00 4 02265          LXA     RCX,4           AND INDEX REGISTERS
02374  0534 00 2 02266          LXA     RCY,2
02375  0534 00 1 02267          LXA     RCZ,1
02376  0601 00 0 01524          STO     $ERAC           SAVE THE CONTENTS OF THE AC
02377 -0754 00 0 00000          PXD     0,0
02400                           ERROR   NOROOM          NOT ENOUGH WORDS COLLECTED
                        *
                        * RELOC        RELOCATES ALL ITEMS IN FULL WORD SPACE INTO A COMPACTED
                        *              BLOCK TO MAKE BLOCKS OF CONTIGOUS STORAGE AVAILABLE FOR
                        *              ARRAYS.
                        *
02403  0634 00 4 02410  RELOC  SXA     RELX,4          SAVE LINK IR
02404  0074 00 4 02355          TSX     RCERR,4         THIS RPUTINE HAS NOT BEEN CODED YET.
02405  454660512543             BCI     3,NO RELOCATOR
02410  0774 00 4 00000  RELX   AXT     **,4            RESTORE LINK IR
02411  0020 00 4 00001          TRA     1,4             RETURN TO MAIN PROGRAM
                        *
                        * MESSAGES AND CONSTANTS PLUS STORAGE GO HERE
                        *
02412  602723602163     RCTM   BCI     1, GC AT
02413  0 00000 0 00000  RCT1                           THE CALL LOCATION IS PUT HERE
02414  606026644343             BCI     2,  FULL WORDS
02416  0 00000 0 00000  RCT4                           NUMBER FULL WORDS COLLECTED
02417  602651252560             BCI     1, FREE
02420  0 00000 0 00000  RCT5                           FREE STORAGE WORDS COLLECTED
02421  606047646230             BCI     3,  PUSH DOWN DEPTH
02424  0 00000 0 00000    RCT6                         DEPTH ON PUSH DOWN LIST GOES HERE
02425  0 00000 0 00000  FWC                           COUNT OF AVAILABLE FULL WORDS
02426  0 00000 0 00000  FSC                           COUNT OF AVAILABLE FREE-STORAGE WORDS
                  02424  GCPDLC SYN     RCT6
02427  0 00000 0 00000    RCC                         TOTAL NUMBER OF RECLAMATION CYCLES
02430  0 00000 0 00000    RCT                         TEST CELL TO SEE IF RELOCATION WAS DON
                  02430  RCRLOC SYN     RCT
02431  0 00000 0 00000    RLC                         NUMBER OF TIMES RELOCATION WAS DONE
02432  0 00000 0 00000    TFWC                        TOTAL FULL WORDS COLLECTED
02433  0 00000 0 00000    TFSC                        TOTAL FREE STORAGE COLLECTED
                  00140    MONES SYN     SEVENS
                  00140    MONS  SYN     SEVENS
02434  0 00000 0 00000    RCIND                       INDICATOR STORAGE
02435  0 00000 0 00000    RCBE                        TEST CELL FOR BAD EXIT
02436  0 00000 0 00000  MARYT                         TEMPORARY STORAGE
                  00074  CRITWN SYN     $Q10
                        *
                        * BIT TABLES FOR MARKING AND SWEEPING FULL WORD SPACE
                        *
02437 +000000000020             OCT     20
02440 +000000000040             OCT     40,100,200,400,1000,2000,4000,10000,20000,40000,100000
02453 +000000200000             OCT     200000,400000,1000000,2000000,4000000,10000000,20000000
02462 +000040000000             OCT     40000000,100000000,200000000,400000000,1000000000
02467 +002000000000             OCT     2000000000,4000000000,10000000000,20000000000
02473 +040000000000             OCT     40000000000,100000000000,200000000000
02476 -000000000000    BIT      OCT     400000000000
                  02477  MBIT   SYN     BIT+1
                  02436  MBITF  SYN     BIT-32
```

ERROR PROCESSING

```
                                *
                                *
02477  0 00000 0 00000  VERBOS PZE                    THIS CEELL NONZERO MAKES GC
                                *                       VERY TALKATIVE
02500 +111111111111     RCSGNL OCT      111111111111
02501 +222222222222     RCSGNM OCT      222222222222
02502 +333333333333     RCSGNN OCT      333333333333    PHASE SIGNALS FOR MC
02503  0 00000 0 00000   TMLM                           TEMPORARY STORAGE
02504 -3 00000 7 00000   TMPTM SVN      ,7              PREFIX AND TAG MASK
02505  0 00000 0 00000   MRKP                           TEMPORARY STORAGE
                                *
02506  0 75270 0 75271   TEMXX          -*-1,,-*-2      PERMENANT TEMLIS ITEMS
02507  0 00176 0 00141                  BCONAT,,ECONAT
02510  0 75266 0 75267                  -*-1,,-*-2
02511  0 15177 0 15171                  C$PROBE,,C$PROEN LAP PROTECTED AREA
02512  0 00000 0 75265                  -*-1            END OF TEMLIS
02513  0 02673 0 02514                  BEGBLK,,ENDBLK-1 FUNCTION STORAGE
```

ERROR PROCESSING

```
                                    INSERT  $TEMLIF
                                *           STORAGE BLOCK FOR FUNCTIONS ALL OVER THE PACKAGE
                                *
02514                           BEGBLK BSS      0
                                *           RECLAIMER STORAGE TO BE MARKED
02514  0 75272 0 00000          TEMLIS          ,,-TEMXX
02515  0 00000 0 00000          ARYLIS                          LIST OF ACTIVE ARRAYS
02516  0 00000 0 00000           RCAC                           AC STORAGE
02517  0 00000 0 00000           RCMQ                           MQ-STORAGE
                                *           CNSFWL STORAGE
02520  0 00000 0 00000           CNXT                           POINTER TO NEXT WORD ON LINEAR OBJLIST
02521  0 00000 0 00000           CNX                            POINTER TO NEXT WORD ON PROPERTY LIST
02522  0 00000 0 00000           CNFT                           POINTER TO NEXT WORD  CN PNAME LIST
02523  0 00000 0 02523           CNAT            *              POINTER TO FIRST WORD OF CURRENT ATOM
02524  0 00000 0 00000           CNVA                           POINTER TO FIRST WORD OF PNAME LIST
                                ************************************************************
                                TEMS    MACRO   LABEL,HEADER,ITEMS MACRO FOR TEDIOUS, UNCOMMENTED PZE'S
                                        HEAD    HEADER
                                TEMTEM  SET     0               ITEM COUNTER, SINCE FIRST IS SPECIAL
                                        IRP     ITEMS
                                TEMTEM  SET     TEMTEM+1        COUNT THIS ITEM
                                        IFF     TEMTEM-1,X      IS THIS THE FIRST
                                        TEMSA   LABEL,ITEMS     *YES
                                        IFF     TEMTEM-1        ONCE AGAIN, IS THIS THE FIRST
                                ITEMS                           *NO
                                        IRP
                                TEMS    END

                                TEMSA   MACRO   LABEL,ITEM      AUXILIARY MACRO FOR FIRST PZE
                                        IFF     0,LABEL,NONAME WAS THE LABEL ARG 'NONAME'
                                ITEM            LABEL           *NO, LABEL FIRST ENTRY
                                        IFF     1,LABEL,NONAME WAS THE LABEL ARG 'NONAME'
                                ITEM                            *YES, DON'T LABEL FIRST ENTRY
                                TEMSA   END

02525                                   TEMS    NONAME,A,CSV  USED SOMEWHERE IN APPLY
02526                                   TEMS    NONAME,0,(ALIST,ARG1,ARG2,ARG3,ARG4,ARG5,ARG6,ARG7,ARG8)
02537                                   TEMS    NONAME,0,(ARG9,ARG10,ARG11,ARG12,ARG13,ARG14,ARG15)
02546                                   TEMS    NONAME,0,(ARG16,ARG17,ARG18,ARG19,ARG20)
02553                                   TEMS    $AND,R,(EVA1,EVA2,EVA9) AND
02556                                   TEMS    $F1,A,(AS1,CWR1) APPEND
02560                                   TEMS    NONAME,A,(ASS1,ASSL,ASSA,AST1,AST2,AST3,AST4)    APPLY
02567                                   TEMS    $COPYN,R,(CS1,CS2) COPY
02571                                   TEMS    NONAME,C,CPF  CP1
02572                                   TEMS    $COND,A,(ECS1,ECS2,ECS3,ECS4) EVCON
                                        HEAD    A                       EVP26
02576  0 00000 0 00000   EVS1                                   IR4, BOTTOM OF PROTECTED TEMP. STORAGE
02577  0 00000 0 00000   EVSE
02600  0 00000 0 00000   EVSA
02601 -0 00000 0 00000   EVTRK MZE                              TRACE SWITCH
02602  0 00000 0 00000   EVTDE                                  CDR(E)
02603  0 00000 0 00000   EVD2
                                        HEAD    R               GO SPECIAL FORM
```

ERRUR PROCESSING

```
 02604  0 00000 0 11760   GOX           $GO                    LINK IR
 02605                          TEMS    NUNAME,R,BFS4 LABP
 02606                          TEMS    NUNAME,R,(BFS2,BFS3) LAMP
                                HEAD    C                      LINK FOR COMPILED FUNCTIONS
 02610  0 00000 0 00000   LNKA                                 LINK STORAGE FOR AC
 02611  0 00000 0 00000   LNKB                                 LINK STORAGE FOR MQ
 02612                          TEMS    $PMAPCA,D,(RET,L,F) MAPCAR
 02615                          TEMS    $R69B,R,(MCN5,MCN4,MCN3,MCN2) MAPCON
                                HEAD    R                      MAPLIS
 02621  0 00000 0 11537   MS1           $R69A                  LINK IR STURAGE
 02622  0 00000 0 00000   MS2                              ARGUMENT L
 02623  0 00000 0 00000   MS3                              FUNCTIONAL ARGUMENT
 02624  0 00000 0 00000   MS4                              FINAL ANSWER
 02625  0 00000 0 00000   MS5                              INTERMEDIATE ANSWER
 02626                          TEMS    $OR,R,(EVR1,EVR2,EVR9) OR
                                HEAD    A                      PAIR
 02631  0 00000 0 00000   TEM                                  FIRST ARGUMENT
 02632  0 00000 0 00000   LIS                                  SECOND ARGUMENT
                        **** PRINAR HAS BEEN DELETED****
                                HEAD    R                      PROGRAM FEATURE
 02633  0 00000 0 11155   INTRX         $PROG                   LINK INDEX REGISTER
 02634  0 00000 0 00000   INTB                                 CURRENT STATEMENT
 02635  0 00000 0 00000   INTGL                                GO LIST,(LIST OF PROGRAM POINTS) + IR2
 02636  0 00000 0 00000   INTPL                                PAIR LIST
 02637  0 00000 0 00000   INTGS                                GO SWITCH , NCN-ZERC IF GC OR RETURN
                                HEAD    I                      READ1
 02640  0 00000 0 11043   RS1           $F13
 02641  0 00000 0 00000   RS2
 02642  0 00000 0 00000   PRINTL                               TEMPORARY STORAGE FCR PRINT CR PUNCH
                                HEAD    R                      SEARCH
 02643  0 00000 0 10626   SRS1          $SRCH                  IR4
 02644  0 00000 0 00000   SRS2                                 L
 02645  0 00000 0 00000   SRS3                                 P
 02646  0 00000 0 00000   SRS4                                 F
 02647  0 00000 0 00000   SRS5                                 U
 02650                          TEMS    $SETQ,R,(REPS1,REPV,REPT1) SETQP
 02653                          TEMS    NONAME,R,(SX1,SZ,SX,SY,S1) SUBST
                                HEAD    Q                      ADD, ETC.
 02660  0 00000 0 00000   AMIR                                 IR 4 STORAGE
 02661  0 00000 0 00000   AMIND                                INDICATOR REGISTER STORAHE
 02662  0 00000 0 00000   AMLIS                                LIST STORAGE
 02663  0 00000 0 00000   AMQ                                  TYPE STORAGE
                        *                                      ARRAY MAKE PROGRAM
 02664  0 00000 0 00000   AFAT                                 ARRAY ATOM GOES HERE
 02665  0 00000 0 00000   ATMP                                 TEMPORARY STORAGE
                                HEAD    S                      EVALQUOTE STORAGE
 02666  0 00000 0 00000   EVQAN
                        ****EVALQUOTE BUFFER HAS BEEN DELETED ,WILL USE FREE STORAGE FOR BUFFER
 02667  0 00000 0 00000   EVQLS PZE
 02670 -0 00000 0 00000   EVQB  MZE                            TEST CELL FOR READ IN
                                HEAD    F
                        *               CHARACTER FUNCTIONS
 02671  0 00000 0 00000   BBPNT PZE                        POINTER TO REMAINDER OF LIST
                        *               MKNO
 02672  0 00000 0 00000   MKT1                                 TEMP STORAGE TYPE (FIX OR FLO)
```

ERROR PROCESSING

```
                    * ROOM FOR MORE STORAGE IN TEMLIS HAS BEEN DELETED****
02673                       TEMS    NONAME,0,XPLE EXPLODE
02674               ENDBLK  BSS     0
                            INSERT  $CONSZF
```

ERROR PROCESSING

```
                                    HEAD    0
                            * CONSW         PUTS FULL WORDS IN FULL WORD SPACE

02674  0634 00 4 02710   CONSW SXA     CSWX,4              SAVE LINK IR
02675 -0534 00 4 02713   FWLOR LXD     FWORDL,4            PICK UP FULL WORD LIST
02676 -3 00000 4 03031         TXL     FWLOUT,4,0          TEST FOR NC MORE
02677 -0600 00 0 02712         STQ     CSWQ                SAVE MQ
02700  0560 00 4 00000         LDQ     0,4                 PICK UP POINTER TO NEXT WORD ON FWL
02701 -0620 00 0 02713         SLQ     FWORDL              UP DATE FULL WORD LIST POINTER
02702  0601 00 4 00000         STO     0,4                 PUT AC IN FULL WORD AREA
02703 -0754 00 4 00000         PXD     0,4                 POINTER TO AC
02704 -0534 00 4 02713         LXD     FWORDL,4            POINTER TO NEXT AVAILABLE WORD
02705  3 00000 4 02707         TXH     CSWO,4,**           BOTTOM FULL WORD SPACE, TEST FOR ARY
02706 -0634 00 4 02705         SXD     *-1,4               AVAILABLE LOCATION AND UPDATE SAME
02707  0560 00 0 02712   CSWO  LDQ     CSWQ                RESTORE MQ
02710  0774 00 4 00000   CSWX  AXT     **,4                RESTORE LINK IR
02711  0020 00 4 00001         TRA     1,4                 EXIT
02712  0 00000 0 00000   CSWQ                             TEMPORARY STORAGE FOR MQ
02713  0 00000 0 00000   FWORDL                           POINTER TO FULL WORD LIST

                            * CONS          BASIC LISP FUNCTION PUTS A WORD IN FREE STORAGE

02714  0634 00 4 02733   CONS  SXA     CNSX,4              SAVE LINK IR
02715 -0534 00 4 02735         LXD     $FREE,4             GET FREE STORAGE LIST POINTER
02716  3 00000 4 02720         TXH     *+2,4,0             SKIP IF NOT OUT OF FREE STORAGE
02717  0074 00 4 03037         TSX     FROUT,4             OUT OF FREE STORAGE
02720  0771 00 0 00022         ARS     18                  DECREMENT TO ADDRESS
02721  0621 00 4 00000         STA     0,4                 PUT ADDRESS AWY
02722  0500 00 4 00000         CLA     0,4                 GET POINTER TO NEXT WORD IN FREE
02723  0622 00 0 02735         STD     FREE                PUT IN FREE
02724 -0620 00 4 00000         SLQ     0,4                 PUT DECREMENT AWAY
02725 -0754 00 4 00000         PXD     0,4                 POINTER TO WORD
02726  0774 00 4 00000   CNTR1 AXT     **,4                LOW ORDER 15 BITS OF CONS COUNTER KEPT
02727  2 00001 4 02732         TIX     *+3,4,1       CONS COUNTER REACTIVATED*******
02730  0074 00 4 02736         TSX     ARREST,4            COUNT EXHAUSTED, RELOAD OR STOP
02731  0774 00 4 77777         AXT     -1,4                RELOAD NUMBER
02732  0634 00 4 02726         SXA     CNTR1,4             PUT IN COUNTER
02733  0774 00 4 00000   CNSX  AXT     **,4                RESTORE LINK IR
02734  0020 00 4 00001         TRA     1,4                 EXIT
02735  0 00000 0 00000   FREE                             POINTER TO FREE STORAGE LIST

02736 -0520 00 0 10447   ARREST NZT     TCOUNT              SKIP IF COUNS COUNTER ON
02737  0020 00 4 00001         TRA     1,4                 OTERWISE RETURN
02740  0601 00 0 03074         STO     CNTM                SAVE AC
02741  0500 00 0 03073         CLA     CNTS                GET REST OF COUNTER
02742  0100 00 0 02747         TZE     AWHOA               GO TO ERROR CALL IF EXHAUSTED
02743  0402 00 0 03075         SUB     CTG                 DECREMENT BY 32,768
02744  0601 00 0 03073         STO     CNTS                UPDATE COUNTER
02745  0500 00 0 03074         CLA     CNTM                RESTORE AC
02746  0020 00 4 00001         TRA     1,4                 EXIT TO RELOAD CNTR1

02747  0634 00 0 10447   AWHOA SXA     TCOUNT,0            DEACTIVATE THE CONS COUNTER
02750  0500 00 0 02754         CLA     CNTPTR        LOCAL -ATOM- TO PRINT CONS COUNT
02751                           ERROR   CNSCTR        CONS COUNTER TRAP
```

ERROR PROCESSING

```
02754   0 75023 0 00000  CNTPTR          ,,-*-1          POINTER TO CONS-COUNT ATOM
02755  -0 03065 1 77777         MZE      -1,1,CNTST      HERE IS THE ATOM

                        * SPEAK          TURNS THE CONTENTS OF THE CONS COUNTER INTO A FIXED POINT
                        *                NUMBER.

02756   0500 00 0 00132   SPEAK CLA      $AMASK          GET ADDRESS MASK
02757  -0320 00 0 02726         ANA      CNTR1           PICK UP 15 LOW ORDER BITS
02760  -0501 00 0 03073         ORA      CNTS            OR IN REST OF COUNT
02761   0601 00 0 03074         STO      CNTM            SAVE CURRENT VALUE
02762   0500 00 0 03065         CLA      CNTST           PICK UP INITIAL VALUE
02763   0402 00 0 03074         SUB      CNTM            SUBTRACT CURRENT VALUE TO GET NUMBER
02764   0560 00 0 00142         LDQ      $FIXD           OF CONSES.  PUT $FIX IN MQ
02765   0020 00 0 11163         TRA      $MKNO           MAKE THE RESULT A NUMBER
```

```
                                        INSERT  $BLCKF2
                                * BLOCKR        BLOCK RESERVATION ROUTINE USED IN DECLARING ARRAYS.
                                *
02766   0634 00 4 03016  BLOCKR SXA     BLKX,4          SAVE LINK IR
02767   0621 00 0 03012          STA     BLKB            BE RESERVED
02770  -0520 00 0 01712          NZT     BPLACE          TEST IF  BPS OR EXCISABLE AREA
02771   0534 00 4 01704          LXA     $ORG,4
02772   0520 00 0 01712          ZET     BPLACE          TEST IF  BPS OR EXCISABLE AREA
02773   0534 00 4 01673          LXA     $XORG,4
02774   0754 00 4 00000          PXA     0,4             ADDRESS OF FIRST REGISTER FOR ARRAYS
02775   0401 00 0 03012          ADM     BLKB            ADDRESS OF END OF BLOCK
02776   0621 00 0 03013          STA     BLKC            INITIALIZE STZ LOOP TO CLEAN OUT BLOCK
02777   0734 00 4 00000          PAX     0,4
03000   0402 00 0 15245          SUB     =1
03001   0621 00 0 03020          STA     BLKBB
03002  -0520 00 0 01712          NZT     BPLACE          CHECK IF OUT OF BPS
03003   0522 00 0 03022          XEC     C$LBPTP
03004   0520 00 0 01712          ZET     BPLACE
03005   0522 00 0 03021          XEC     LTBPFJ
03006  -0520 00 0 01712          NZT     BPLACE          TEST IF  BPS OR EXCISABLE AREA
03007   0634 00 4 01704          SXA     $ORG,4          UPDATE ORG
03010   0520 00 0 01712          ZET     BPLACE          TEST IF  BPS OR EXCISABLE AREA
03011   0634 00 4 01673          SXA     $XORG,4
03012   0774 00 4 00000  BLKB    AXT     **,4            LENGTH OF BLOCK
03013   0600 00 4 00000  BLKC    STZ     **,4            ZERO THE BLOCK
03014   2 00001 4 03013          TIX     *-1,4,1
03015   0500 00 0 03020          CLA     BLKBB           GET ANSWER
03016   0774 00 4 00000  BLKX    AXT     **,4            RESTORE LINK IR
03017   0020 00 4 00001          TRA     1,4
03020   0 00000 0 00000  BLKBB                           ANSWER STORED HERE
                                *
                                * LBPTP CHECKS FOR OUT OF BPS AND MAKES ERROR IF D SU.
                                *       LTBPFJ  CHECKS IF OUT OF EXCISABLE SPACE
                                *
                                        HEAD    C                       FOR LAP
03021   3 77161 4 03023  LTBPFJ TXH     **+2,4,LAP-1
03022   3 00000 4 03023  LBPTP  TXH     **+1,4,**       SETUP FILLS THIS CELL
03023  -0634 00 4 01530          SXD     $ERROR,4
03024  -0754 00 4 00000          PXD     0,4
03025   0560 00 0 00150          LDQ     $OCTD
03026   0074 00 4 11163          TSX     $MKNO,4
03027   0074 00 4 01531          TSX     $ERROR+1,4
03030   224762466463             BCI     1,BPSOUT
                                *
                                *  JUST REDUCES THE AC MOD 2**15.THE RESULT IS 15 BITS IN ADDR OF AC
                                        HEAD    0
                                *                VARIOUS ENTRANCES TO THE RECLAIMER
                                *
                                *                FWLOUT - OUT OF FULL WORD LIST
03031   0601 00 0 02712  FWLOUT STO     CSWQ            SAVE FULL WORD
03032  -0754 00 0 00000          PXD     0,0             ZERO AC
03033   0600 00 0 02430          STZ     RCRLOC          SIGNAL NO RELOCATION IS NECESSARY
03034   0074 00 4 02007          TSX     RECLAM,4        DO THE WORK
03035   0500 00 0 02712          CLA     CSWQ            RESTORE AC
03036   0020 00 0 02675          TRA     FWLOR           RETURN TO CONSW
```

```
                                 *            FROUT - OUT OF REE STORAGE
03037  0634 00 4 03042   FROUT SXA   FRX,4                SAVE LINK IR
03040  0600 00 0 02430         STZ   RCRLOC               SIGNAL NO RELOCATION NECESSARY
03041  0074 00 4 02007         TSX   RECLAM,4             DO THE WORK
03042  0774 00 4 00000   FRX   AXT   **,4                 RESTORE LINK OR
03043  0020 00 4 77776         TRA   -2,4                 NON-STANDARD EXIT
                                 *
                                 *
                                 * COUNT      A FUNCTION OF 1 ARGUMENT (A FIXED POINT NUMBER) TURNS ON
                                 *            THE CONS COUNTER AND LOADS IT WITH THAT NUMBER
                                 *            A LOAD OF NIL  SIMPLY LEAVES THE PREVIOUS CONTENTS IN THE
                                 *            COUNTER
                                 *
03044 -0625 00 0 10447   COUNT STL   TCOUNT               ACTIVATE THE CONS COUNTER
03045 -0100 00 0 03051         TNZ   CNTA                 GO IF ARUGMENT IS NOT NULL
03046  0500 00 0 03074         CLA   CNTM                 OLD VALUE OF CNTR1
03047  0621 00 0 02726         STA   CNTR1                PUT IT THERE
03050  0020 00 0 03063         TRA   CNTB                 CLEAR AC AND EXIT
03051  0634 00 4 03061   CNTA  SXA   CNTX,4               RELOAD COUNTER WITH FIXED POINT ARG.
03052  0634 00 2 03062         SXA   CNTY,2               SAVE INDEX REGISTERS
03053 -0734 00 2 00000         PDX   0,2                  ARGUMENT TO INDEX 2
03054  0074 00 4 11423         TSX   FIXVAL,4             EVALUATE AS A FIXED POINT NUMBER
03055  0601 00 0 03065         STO   CNTST                SET INITIAL VALUE CELL
03056  0621 00 0 02726         STA   CNTR1                LOW ORDER 15 BITS TO CNTR1
03057 -0320 00 0 00135         ANA   PDTMSK               MASK OUT LOW ORDER 15 BITS
03060  0601 00 0 03073         STO   CNTS                 STORE REMAINDER IN HIGH ORDER CELL
03061  0774 00 4 00000   CNTX  AXT   **,4                 RESTORE INDEX REGISTERS
03062  0774 00 2 00000   CNTY  AXT   **,2
03063 -0754 00 0 00000   CNTB  PXD   0,0                  GIVE VALUE OF NIL
03064  0020 00 4 00001         TRA   1,4                  EXIT
03065  0 00000 0 00000   CNTST                            INTAL VALUE OF COUNT
                                 *
                                 * UNCONT     DEACTIVATES THE CONS COUNTER
                                 *
03066  0634 00 0 10447  UNCONT SXA   TCOUNT,0             DEACTIVATE THE CONS COUNTER
03067  0500 00 0 02726         CLA   CNTR1                GET CURENT CONTENST OF COUNTER
03070  0621 00 0 03074         STA   CNTM                 SAVE IN TEMP STORAGE
03071 -0754 00 0 00000         PXD   0,0                  GIVE VALUE OF NULL
03072  0020 00 4 00001         TRA   1,4                  EXIT
                                 *
03073  0 00000 0 00000   CNTS                             HIGH ORDER BITS OF CONS COUNTER
03074  0 00000 0 00000   CNTM                             TEMPORARY STORAGE
03075  0 00000 1 00000   CTG         ,1                   LOW ORDER BIT OF HIGH ORDER 20 BITS
                               INSERT DECONF
```

```
                        *
                        E       HED
                        *       DECON AND NUMNAM
                        *
                        *       DECON TAKES A DECIMAL INTEGER (+ OR -) AS INPUT IN THE AC AND
                        *       GIVES AS OUTPUT THE BCD REPRESENTATION OF THAT NUMBER.  LO ORDER
                        *       BITS ARE IN AC, HI ORDER BITS IN MQ.  LEADING ZEROS ARE
                        *       SUPPRESSED.  IF THERE ARE NO HI ORDER BITS, MQ IS ZERO.  THE
                        *       P BIT AND SIGN OF AC WILL AGREE.
                        *
                        *       NUMNAM TAKES AS INPUT A POINTER TO A DECIMAL INTEGER (+ OR -) AND
                        *       CAUSES THE BCD REPRESENTATION OF THAT NUMBER TO BE PRINTED, WITH
                        *       LEADING ZEROS SUPPRESSED.

03076   0600 00 0 00044   DECON STZ     DETS1           SIGNAL FOR DECON EXIT
03077   0600 00 0 00046         STZ     DELOD           SET LO ORDER DIGITS TO ZERO
03100   0634 00 4 03163         SXA     DEIR4,4         SAVE IR4
03101   0020 00 0 03106         TRA     DE7

03102  -0625 00 0 00044   NUMNAM STL     DETS1           SIGNAL FOR NUMNAM EXIT
03103   0634 00 4 03163         SXA     DEIR4,4         SAVE IR4
03104  -0734 00 4 00000         PDX     ,4              PLACE INPUT NUMBER IN AC
03105   0500 00 4 00000         CLA     0,4
03106  -0625 00 0 00045   DE7   STL     DETS2           SIGNAL FOR NO HI- ORDER DIGITS
03107   0601 00 0 00051         STO     DEINP           SAVE INPUT FOR SIGN TEST
03110   0760 00 0 00012         DCT                     SHUT OFF DIVIDE CHECK LIGHT
03111   0761 00 0 00000         NOP
03112  -0130 00 0 00000         XCL                     NUMBER TO MQ
03113   0774 00 4 00044         AXT     36,4            INDEX FOR SHIFTING
03114   0600 00 0 00047   DE4   STZ     DEDIG           DEDIG WILL RECIEVE DIGITS

03115  -0754 00 0 00000   DE1   PXD     ,0
03116   0221 00 0 00074         DVP     $Q10            PUT ANOTHER DIGIT IN DEDIG
03117   0767 00 4 00044         ALS     36,4
03120  -0602 00 0 00047         ORS     DEDIG
03121  -0600 00 0 00050         STQ     DEMQ            IF QUOTIENT ZERO, CONVERSION
03122  -0520 00 0 00050         NZT     DEMQ              IS DONE
03123   0020 00 0 03131         TRA     DE2
03124   2 00006 4 03115         TIX     DE1,4,6

03125   0500 00 0 00047         CLA     DEDIG           STORE LO ORDER DIGITS
03126   0601 00 0 00046         STO     DELOD
03127   0600 00 0 00045         STZ     DETS2           SIGNAL THAT HI ORDER DIGITS EXIST
03130   1 00036 4 03114         TXI     DE4,4,30        RESTORE SHIFT INDEX AND LOOP AGAIN

03131   0560 00 0 00051   DE2   LDQ     DEINP           SEE IF MINUS SIGN NEEDED
03132   0162 00 0 03144         TQP     DEV
03133   2 00006 4 03141         TIX     DEQ,4,6

                        *       MINUS SIGN BEGINS A NEW WORD
03134   0500 00 0 00047         CLA     DEDIG           STORE LO ORDER DIGITS
03135   0601 00 0 00046         STO     DELOD
03136   0600 00 0 00045         STZ     DETS2           SIGNAL THAT HI ORDER DIGITS EXIST
03137   0600 00 0 00047         STZ     DEDIG           CLEAR DIGITS REGISTER
```

```
03140   0774 00 4 00044          AXT     36,4            RESTORE SHIFT INDEX
03141   0500 00 0 15264   DEQ    CLA     =H00000-        INSERT MINUS SIGN
03142   0767 00 4 00044          ALS     36,4
03143  -0602 00 0 00047          ORS     DEDIG

03144   0760 00 0 00012   DEV    DCT
03145   0074 00 4 01624          TSX     $DCT,4          MACHINE ERROR
03146   0520 00 0 00044          ZET     DETS1           SEE WHICH EXIT TO USE
03147   0020 00 0 03165          TRA     DE5

                          *      DECON EXIT
03150  -0500 00 0 00047          CAL     DEDIG           PICK UP DIGITS
03151  -3 00006 4 03155          TXL     DEJ,4,6         TRANSFER IF FULL WORD OF DIGITS
03152  -0765 00 4 00052          LGR     42,4            INSERT LEADING BLANKS
03153  -0500 00 0 15311          CAL     =H
03154  -0763 00 4 00052          LGL     42,4
03155   0560 00 0 00046   DEJ    LDQ     DELOD           LO ORDER DIGITS OR ZERO -
03156  -0520 00 0 00045          NZT     DETS2             SEE WHICH
03157  -0130 00 0 00000          XCL                     LO ORDER DIGITS TO AC
03160  -0760 00 0 00001          PBT                     SIGN AND P BIT MUST AGREE
03161   0020 00 0 03163          TRA     *+2
03162  -0760 00 0 00003          SSM
03163   0774 00 4 00000   DEIR4  AXT     **,4            RESTORE IR4 AND EXIT
03164   0020 00 4 00001          TRA     1,4

                          *      NUMNAM EXIT
03165  -0500 00 0 00047   DE5    CAL     DEDIG           INSERT TRAILING SEVENS INTO
03166   0560 00 0 00140          LDQ     SEVENS            DIGITS WORD
03167  -0765 00 4 00052          LGR     42,4
03170   0131 00 0 00000          XCA
03171   0074 00 4 04076          TSX     $PRIN2,4        PRINT WORD OF DIGITS
03172   0520 00 0 00045          ZET     DETS2           SEE IF ANOTHER WORD MUST
03173   0020 00 0 03176          TRA     DEY               BE PRINTED
03174  -0500 00 0 00046          CAL     DELOD           PRINT LO ORDER DIGITS
03175   0074 00 4 04076          TSX     $PRIN2,4

03176   0534 00 4 03163   DEY    LXA     DEIR4,4         RESTORE IR4,  CLEAR AC, AND EXIT
03177  -0754 00 0 00000          PXD     ,0
03200   0020 00 4 00001          TRA     1,4

03201                     DEORG  BSS     0
                    00044         ORG     COMMON
00044                     DETS1  BSS     1               ZERO MEANS DECON EXIT
00045                     DETS2  BSS     1               ZERO MEANS HI ORDER DIGITS
00046                     DELOD  BSS     1               LO ORDER DIGITS
00047                     DEDIG  BSS     1               CURRENT DIGITS
00050                     DEMQ   BSS     1               MQ FOR ZERO TEST
00051                     DEINP  BSS     1               INPUT NUMBER
                    03201         ORG     DEORG
                                  INSERT  $EQULZF
```

```
                        *       THIS ROUTINE USES COMMON, SEVENS, $PRIN2, BLANKS, AND $QIO
                        *
                        R       HED
                                        MAPLIS          NEW, FASTER VERSION WITH OPEN SAVE AND CONS
                        *
03201  0100 00 4 00001  MAPLIS TZE     1,4             NULL(L) = NIL
03202 -0634 00 4 02621          SXD     MS1,4           SAVE LINK IR
03203 -0534 00 4 01727          LXD     $CPPI,4         GET PDL POINTER
03204  1 77772 4 03205          TXI     *+1,4,-6                SAVE TOTAL OF 6 ITEMS
03205  0522 00 0 02003          XEC     $ENDPDL                 TEST FOR OUT OF PUSH DOWN LIST
03206 -0634 00 4 01727          SXD     $CPPI,4         UPDATE PDL POINTER LOCATION
03207 -0534 00 1 01727          LXD     $CPPI,1                 MAKE IX1 AND $CPPI AGREE
03210  0601 00 0 02527          STO     $ARG1           SAVE AC
03211  0500 00 0 02621          CLA     MS1             START SAVING    LINK IR
03212  0601 00 4 77772          STO     -6,4
03213  0500 00 0 02622          CLA     MS2             L ARGUMENT
03214  0601 00 4 77773          STO     -5,4
03215  0500 00 0 02623          CLA     MS3             FUNCTIONAL ARGUMENT
03216  0601 00 4 77774          STO     -4,4
03217  0500 00 0 02624          CLA     MS4             FINAL ANSWER
03220  0601 00 4 77775          STO     -3,4
03221  0500 00 0 02625          CLA     MS5             INTERMEDIATE ANSWER
03222  0601 00 4 77776          STO     -2,4
03223  0500 00 0 03333          CLA     MS6                     SAVE MARKER
03224  0601 00 4 77777          STO     -1,4
03225  0500 00 0 02527          CLA     $ARG1           SAVING ALL DONE, RESTORE AC
03226  0601 00 0 02622          STO     MS2             PUT L ARGUMENT AWAY
03227 -0600 00 0 02623          STQ     MS3             PUT FUNCTIONAL ARGUMENT AWAY
03230  0162 00 0 03323          TQP     CMP             IF TRANSFER, F NOT A TXL, SO GO TO COMPAT
03231  0074 00 4 02623          TSX     MS3,4           EXECUTE FUNCTIONAL ARGUMENT
03232 -0534 00 4 02735  MAIN    LXD     $FREE,4         START OPEN CONS
03233  3 00000 4 03235          TXH     *+2,4,0         TEST FOR OUT OF FREE STORAGE
03234  0074 00 4 03037          TSX     $FROUT,4        GO IF NO MORE FS
03235  0771 00 0 00022          ARS     18              PUT F(L) IN ADDRESS
03236  0560 00 4 00000          LDQ     0,4             GET NEXT REGISTER ON FSL
03237 -0620 00 0 02735          SLQ     $FREE           UPDATE FREE
03240  0601 00 4 00000          STO     0,4             CONS(F(L),NIL)
03241 -0634 00 4 02624          SXD     MS4,4           FINAL ANSWER
03242 -0634 00 4 02625          SXD     MS5,4           INT. ANSWER
03243  0534 00 4 02726          LXA     $CNTR1,4                PICK UP CONS COUNTER
03244  2 00001 4 03247          TIX     *+3,4,1                 DECREMENT BY 1
03245  0074 00 4 02736          TSX     ARREST,4                GO IF OUT OF COUNTER
03246  0774 00 4 77777          AXT     -1,4                    RELOAD OF -1 FOR COUNTER
03247  0634 00 4 02726          SXA     $CNTR1,4                RESTORE CONS COUNTER
03250 -0534 00 4 02622  MLOP1   LXD     MS2,4           MAUN LOOP,  GET L
03251  0500 00 4 00000          CLA     0,4             TAKE CDR(L)
03252 -0734 00 4 00000          PDX     0,4
03253  3 00000 4 03275          TXH     MPRG1,4,0       IF NOT NULL GO ON TO MAIN PROPRAM
03254  0500 00 0 02624          CLA     MS4             ALL DONE, PICK UP FINAL ANSWER
03255 -0534 00 4 01727          LXD     $CPPI,4         START OPEN UNSAVE BY GETTING PDL POINTER
03256  0560 00 4 77776          LDQ     -2,4
03257 -0600 00 0 02625          STQ     MS5
03260  0560 00 4 77775          LDQ     -3,4
03261 -0600 00 0 02624          STQ     MS4
```

```
03262  0560 00 4 77774        LDQ     -4,4
03263 -0600 00 0 02623        STQ     MS3
03264  0560 00 4 77773        LDQ     -5,4
03265 -0600 00 0 02622        STQ     MS2
03266  0560 00 4 77772        LDQ     -6,4
03267 -0600 00 0 02621        STQ     MS1
03270  1 00006 4 03271        TXI     *+1,4,6         RESTORE PDL COUNTER
03271 -0634 00 4 01727        SXD     $CPPI,4         SET CPPI
03272 -0534 00 4 02621        LXD     MS1,4           PICK UP LINK IR
03273 -0534 00 1 01727        LXD     $CPPI,1         MAKE IX1 AND $CPPI AGREE
03274  0020 00 4 00001        TRA     1,4             RETURN
                       *
03275 -0754 00 4 00000  MPRG1 PXD     0,4             MAIN PROGRAM  PUT L IN AC
03276  0601 00 0 02622        STO     MS2             SAVE IN L ARGUMENT REGISTER
03277 -0534 00 4 02623        LXD     MS3,4           SEE IF FUNCTIONAL ARG IS S EXPRESSION
03300  3 00012 4 03327        TXH     CMP1,4,10       GO IF S EXPRESSION
03301  0074 00 4 02623        TSX     MS3,4           EXECUTE FUNCTIONAL ARGUMENT (TXL INS.)
03302 -0534 00 4 02735  MAIN1 LXD     $FREE,4         START OPEN CONS
03303  3 00000 4 03305        TXH     *+2,4,0         TEST FOR OUT OF FREE STORAGE
03304  0074 00 4 03037        TSX     $FROUT,4        GO IF OUT
03305  0560 00 4 00000        LDQ     0,4             PICK UP POINTER TO NEXT FREE REGISTER
03306 -0620 00 0 02735        SLQ     $FREE           UPDATE FREE
03307  0771 00 0 00022        ARS     18              ITEM TO ADDRESS
03310  0601 00 4 00000        STO     0,4             CONS(F(L),NIL)
03311 -0754 00 4 00000        PXD     0,4             ANSWER TO AC
03312  0534 00 4 02726        LXA     $CNTR1,4        PICK UP CONS COUNTER
03313  2 00001 4 03316        TIX     *+3,4,1         DECREMENT BY 1
03314  0074 00 4 02736        TSX     ARREST,4        GO IF OUT OF COUNTER
03315  0774 00 4 77777        AXT     -1,4            RELOAD OF -1 FOR COUNTER
03316  0634 00 4 02726        SXA     $CNTR1,4        RESTORE CONS COUNTER
03317 -0534 00 4 02625        LXD     MS5,4           PICK UP LAST ANSWER
03320  0622 00 4 00000        STD     0,4             CONCATENATE THE ANSWERS BY RPLACD
03321  0601 00 0 02625        STO     MS5             UPDATE INT. ANSWER
03322  0020 00 0 03250        TRA     MLOP1           GO TO HEAD OF MAIN LOOP
                       *
03323 -0620 00 0 03325  CMP   SLQ     *+2             COMPAT CALL FOR S EXPRESSION FUN. ARG.
03324  0074 00 4 10565        TSX     COMPAT,4
03325  0 00000 0 00001        1,,**           FUNCTION OF 1 ARGUMENT
03326  0020 00 0 03232        TRA     MAIN            GO BACK TO MAIN PROGRAM
                       *
03327 -0634 00 4 03331  CMP1  SXD     *+2,4           ANOTHER COMPAT CALL
03330  0074 00 4 10565        TSX     COMPAT,4
03331  0 00000 0 00001        1,,**
03332  0020 00 0 03302        TRA     MAIN1           RETURN TO MAIN PROGRAM
                       *
03333 -3 02627 0 01760  MS6   TXL     $END5,,MS5+2        SAVE 5 ITEMS

                              FUNCTION COPY
                              COPY(L)= (L=0 YIELDS 0, CAR(L)=-1 YIELDS L,
                                 OTHERWISE CONS(COPY(CAR(L)),COPY(CDR(L))))
                       R      HED
03334  0100 00 4 00001  COPY  TZE 1,4                     L=0
03335 -0634 00 4 02567        SXD CS1,4
03336 -0734 00 4 00000        PDX 0,4                     L
03337 -0634 00 4 03366        SXD CT1,4                   L
```

LISP FUNCTIONS

```
03340  0500 00 4 00000          CLA 0,4                  CWR(L)
03341  0734 00 4 00000          PAX 0,4                  CAR(L)
03342 -3 77776 4 03346          TXL C1,4,-2              CAR(L)=-1
03343  0500 00 0 03366          CLA CT1                  L
03344 -0534 00 4 02567          LXD CS1,4
03345  0020 00 4 00001          TRA 1,4
03346  0074 00 4 01723   C1     TSX $SAVE,4
03347 -3 02572 0 01766          TXL     $END2,,CS2+2     SAVE 2 ITEMS
03350 -0534 00 4 03366          LXD CT1,4                L
03351  0500 00 4 00000          CLA 0,4                  CWR(L)
03352  0601 00 0 02570          STO CS2
03353 -0320 00 0 00133          ANA DECM                 CDR(L)
03354  0074 00 4 03334          TSX COPY,4               COPY(CDR(L))
03355  0534 00 4 02570          LXA CS2,4                CAR(L)
03356  0601 00 0 02570          STO CS2                  COPY(CDR(L))
03357 -0754 00 4 00000          PXD 0,4
03360  0074 00 4 03334          TSX COPY,4               COPY(CAR(L))
03361  0560 00 0 02570          LDQ CS2
03362  0074 00 4 02714          TSX $CONS,4
03363  0074 00 4 01736          TSX UNSAVE,4
03364 -0534 00 4 02567          LXD CS1,4
03365  0020 00 4 00001          TRA 1,4
03366  0 00000 0 00000   CT1
                 00133   DECM   SYN     $DMASK

                                     FUNCTION SEARCH
                                     SEARCH(L,P,F,U)=(L=0 YIELDS U,P(L) YIELDS F(L),
                                         OTHERWISE SEARCH (CDR(L),P,F,U))

                         R      HED
03367 -0634 00 4 02643   SEARCH SXD     SRS1,4
03370  0074 00 4 01723          TSX $SAVE,4
03371 -3 02651 0 01760          TXL     $END5,,SRS5+2    SAVE 5 ITEMS
03372 -0600 00 0 02645          STQ SRS3                 P
03373  0100 00 0 03434   SR3    TZE     SR4
03374  0601 00 0 02644          STO SRS2                 L
03375  0560 00 0 02531          LDQ $ARG3                F
03376 -0600 00 0 02646          STQ SRS4
03377  0560 00 0 02532          LDQ $ARG4                U
03400 -0600 00 0 02647          STQ SRS5
03401 -0534 00 4 02645          LXD SRS3,4
03402  3 00012 4 03405          TXH *+3,4,10
03403  0074 00 4 02645          TSX SRS3,4
03404  0020 00 0 03410          TRA *+4
03405 -0634 00 4 03407          SXD *+2,4
03406  0074 00 4 10565          TSX COMPAT,4
03407  0 00000 0 00001              1,,**
03410  0100 00 0 03424          TZE SR1                  NOT P(L)
03411  0500 00 0 02644          CLA SRS2                 L
03412 -0534 00 4 02646          LXD SRS4,4
03413  3 00012 4 03416          TXH *+3,4,10
03414  0074 00 4 02646          TSX SRS4,4
03415  0020 00 0 03421          TRA *+4
03416 -0634 00 4 03420          SXD *+2,4
03417  0074 00 4 10565          TSX COMPAT,4
```

49304

```
03420   0 00000 0 00001                 1,,**
03421   0074 00 4 01736           TSX UNSAVE,4
03422  -0534 00 4 02643           LXD SRS1,4
03423   0020 00 4 00001           TRA 1,4
03424   0500 00 0 02647   SR1     CLA SRS5                    1 YIELDS
03425   0601 00 0 02532           STO $ARG4                   U
03426   0500 00 0 02646           CLA SRS4
03427   0601 00 0 02531           STO $ARG3                   F
03430  -0534 00 4 02644           LXD SRS2,4                  L
03431   0500 00 4 00000           CLA 0,4
03432  -0320 00 0 00133           ANA DECM                    CDR(L)
03433   0020 00 0 03373           TRA     SR3
03434   0074 00 4 01736   SR4     TSX     UNSAVE,4
03435  -0534 00 4 02532           LXD $ARG4,4
03436   3 00012 4 03441           TXH     SRCMPT,4,10
03437  -0534 00 4 02643           LXD     SRS1,4
03440   0020 00 0 02532           TRA     $ARG4
                          *
03441   0600 00 0 02531   SRCMPT  STZ     $ARG3
03442   0560 00 0 02531           LDQ     $ARG3
03443   0074 00 4 02714           TSX     $CONS,4
03444   0131 00 0 00000           XCA
03445   0500 00 0 02532           CLA     $ARG4
03446  -0534 00 4 02643           LXD     SRS1,4
03447   0020 00 0 13110           TRA     $APPLY
```

```
                          FUNCTION EQUAL
                          EQUAL(L1,L2)=(L1=L2 YIELDS1,L1=0VL2=0 YIELDS 0,
                          CAR(L1)=-1VCAR(L2)=-1 YIELDS 0, OTHERWISE
                               EQUAL(CAR(L1,CAR(L2))AEQUAL(CDR(L1),CDR(L2)))
```

```
                          L       HED
                          * EQUAL         A FUNCTION OF 2 ARGUMENTS DETERMINES WETHER 2 LIST
                          *               STRUCTURES ARE EQUIVELENT.  REPROGRAMMED 5 OCTOBER 1960
                          *               TO MAKE USE OF THE NUMBER CONVENTIONS CURRENTLY IN USE.
                          *
03450  -0634 00 4 03561   EQUAL  SXD     EQXR,4          SAVE LINK IR
03451  -0600 00 0 03563          STQ     EQL2            SAVE ARGUMENT 2
03452   0601 00 0 03562          STO     EQL1            SAVE ARGUMENT 1
03453   0402 00 0 03563   EQLP   SUB     EQL2            EQ TEST
03454   0100 00 0 03505          TZE     EQT             TWO LIST ARE EQ, EXIT TRUE
03455  -0520 00 0 03562          NZT     EQL1            SKIP IF L1 NON NULL
03456   0020 00 0 03510          TRA     EQF             L1 NULL BUT NOT EQ L2, EXIT FALSE
03457  -0520 00 0 03563          NZT     EQL2            NULL TEST L2
03460   0020 00 0 03510          TRA     EQF             L2 NULL BUT NOT EQ  L1, EXIT FALSE
03461  -0534 00 4 03563          LXD     EQL2,4          PICK UP LIST 2
03462   0500 00 4 00000          CLA     0,4             GET NEXT ELEMENT
03463   0734 00 4 00000          PAX     0,4             CAR OF LIST 2
03464   3 77776 4 03513          TXH     EQA,4,-2        GO IF ATOM
03465   0622 00 0 03563          STD     EQL2
03466  -0754 00 4 00000          PXD     0,4             CAR OF LIST TO DECREMENT OF AC
03467   0131 00 0 00000          XCA                     SWITCH TO MQ
03470  -0534 00 4 03562          LXD     EQL1,4          PICK UP LIST 1
03471   0500 00 4 00000          CLA     0,4             GET NEXT ELEMENT
03472   0622 00 0 03562          STD     EQL1            SAVE CDR OF LIST 1
```

```
03473  0734 00 4 00000          PAX     0,4             CAR OF LIST TO IR 4
03474  3 77776 4 03510          TXH     EQF,4,-2        GO TO FALSE EXIT IF THIS IS AN ATUM
03475 -0754 00 4 00000          PXD     0,4             CAR OF LIST TO DECREMENT OF AC
03476  0074 00 4 01723          TSX     $SAVE,4         SAVE CALL
03477 -3 03565 0 01764          TXL     $END3,,EQL2+2   SAVE 3 ITEMS
03500  0074 00 4 03450          TSX     $EQUAL,4        TEST FOR EQUALITY IN CAR  DIRECTION
03501  0074 00 4 01736          TSX     UNSAVE,4        UNSAVE CALL
03502  0100 00 0 03510          TZE     EQF             WHOLE LIST IS FALSE IF CAR DIRECTION F
03503  0500 00 0 03562          CLA     EQL1            PICK UP REST OF LIST 1
03504  0020 00 0 03453          TRA     EQLP            TEST EQUALITY IN CDR DIRECTION
                        *
03505  0500 00 0 00126  EQT     CLA     $QD1            TRUE EXIT, PICK UP 1 IN DECREMENT
03506 -0534 00 4 03561          LXD     EQXR,4          RESTORE LINK IR
03507  0020 00 4 00001          TRA     1,4
                        *
03510 -0754 00 0 00000  EQF     PXD     0,0             FALSE EXIT, CLEAR AC
03511 -0534 00 4 03561          LXD     EQXR,4          RESTORE LINK IR
03512  0020 00 4 00001          TRA     1,4             EXIT
                        *
03513 -0534 00 4 03562  EQA     LXD     EQL1,4          L2 IS ATOM, IS L1
03514  0500 00 4 00000          CLA     0,4
03515  0734 00 4 00000          PAX     0,4
03516 -3 77776 4 03510          TXL     EQF,4,-2
03517 -0534 00 4 03561          LXD     EQXR,4          OTHERWISE DO EQP
03520  0500 00 0 03562          CLA     EQL1
03521  0560 00 0 03563          LDQ     EQL2
03522  0020 00 0 03523          TRA     EQPROG
                        *
                        *       EQP TESTS FOR EQ BETWEEN LISTS AND NUMERICAL EQUALITY BETWEEN
                        *       NUMBERS.   USES A TOLERENCE  IN TESTIONG FLOATION PT NUMBERS
                        *
                        *       DEFINE  (((EQP (LAMBDA (X Y) (COND
                        *                      ((AND (NUMBERP X) (NUMBERP Y)) (ZEROP (DIFFERENCE(X Y)))
                        *                      (T F)  ))) ))
                        *
                        *       IF THIS DEFINITION OF EQP DOES NOT SUIT THE USER,LET HIM
                        *       MAKE HIS OWN USING LISP COMPILER OR LAP.
                        *
03523  0040 00 0 03530  EQPROG TLQ     EQPF
03524  0131 00 0 00000          XCA
03525  0040 00 0 03530          TLQ     EQPF
03526  0500 00 0 00126          CLA     $QD1
03527  0020 00 4 00001          TRA     1,4                     ARE EQ, EXIT TRUE
                        *
03530  0634 00 4 03552  EQPF    SXA     EQPRX,4                 INVESTIGATE FURTHER
03531 -0734 00 4 00000          PDX     0,4
03532  0634 00 2 03553          SXA     EQPRY,2
03533  0500 00 4 00000          CLA     0,4
03534  0734 00 2 00000          PAX     0,2
03535 -3 77776 2 03555          TXL     EQPFX,2,-2
03536 -0320 00 0 00137          ANA     TAGMSK
03537  0100 00 0 03555          TZE     EQPFX
03540 -0754 00 4 00000          PXD     0,4
03541  0131 00 0 00000          XCA
03542 -0734 00 4 00000          PDX     0,4
```

```
03543  0500 00 4 00000          CLA     0,4
03544  0734 00 2 00000          PAX     0,2
03545 -3 77776 2 03555          TXL     EQPFX,2,-2
03546 -0320 00 0 00137          ANA     TAGMSK
03547  0100 00 0 03555          TZE     EQPFX
03550 -0754 00 4 00000          PXD     0,4
03551  0074 00 4 12042          TSX     $DIFFER,4
03552  0774 00 4 00000  EQPRX  AXT     **,4
03553  0774 00 2 00000  EQPRY  AXT     **,2
03554  0020 00 0 12734          TRA     $ZEROP
                        *
03555  0534 00 4 03552  EQPFX  LXA     EQPRX,4
03556  0534 00 2 03553          LXA     EQPRY,2
03557 -0754 00 0 00000          PXD     0,0
03560  0020 00 4 00001          TRA     1,4
                        *
03561  0 00000 0 12376  EQXR           $F8             INDEX REGISTER STORAGE
03562  0 00000 0 00000  EQL1                           LIST 1 STORAGE
03563  0 00000 0 00000  EQL2                           LIST 2 STORAGE
                        *
                        *  EQUAL USES $SAVE,$QD1,UNSAVE,$EQUAL AND FIXFLO
```

```
                        INSERT  $PRNTZF

                        PRINT                   MAY 14,1959

                T       HED

                        PRINT MASTERMINDER

03564  0634 00 4 03573  PRINT SXA     PRPS1,4          SAVE LINK IR
03565 -0534 00 4 01727          LXD     $CPPI,4          SAVE CURRENT CONTENTS OF CPPI
03566 -0634 00 4 04313          SXD     PCPPI,4
03567  0601 00 0 02642          STO     PRINTL          SAVE THE ARGUMENT
03570  0074 00 4 03577          TSX PRIN0,4
03571  0074 00 4 04230          TSX TERPRI,4
03572  0500 00 0 02642          CLA     PRINTL          RESTORE THE ARGUMENT
03573  0774 00 4 00000  PRPS1 AXT     **,4             RESTORE LINK IR
03574  0020 00 4 00001          TRA 1,4
03575  0500 00 0 03655  PRNIL CLA     PRBLW            PICK UP NIL REPRESENTATION
03576  0020 00 0 04076          TRA     $PRIN2          PUT IN PRINT LINE AND EXIT
03577 -0634 00 4 03656  PRIN0 SXD PS1,4
03600  0100 00 0 03575          TZE     PRNIL            PRINT THE NULL LIST
03601 -0734 00 4 00000          PDX 0,4
03602 -0634 00 4 03664          SXD L1,4
03603  0500 00 4 00000          CLA 0,4
03604  0601 00 0 03663          STO CWRL
03605  0734 00 4 00000          PAX 0,4
03606 -3 77776 4 03612          TXL     XA1,4,-2
03607  0500 00 0 03664          CLA L1
03610 -0534 00 4 03656          LXD PS1,4
03611  0020 00 0 03665          TRA $PRIN1
03612 -0534 00 4 04273  XA1     LXD     WORDS,4         ***CHANGE*****
03613  3 00002 4 03615          TXH     *+2,4,2              IF NEAR END OF LINE
03614  0074 00 4 04230          TSX     TERPRI,4             THEN DO A TERPRI
03615  0500 00 0 03661          CLA     LPAR2
03616  0074 00 4 04076          TSX $PRIN2,4
03617  0500 00 0 03663          CLA CWRL
03620  0074 00 4 01723          TSX $SAVE,4
03621 -3 03661 0 01766          TXL     $END2,,PS2+2         SAVE 2 ITEMS
03622  0622 00 0 03657  A3      STD     PS2              SAVE LIST
03623  0734 00 4 00000          PAX     0,4              CAR TO IR 4
03624 -3 00000 4 03651          TXL PRP2,4,0
03625 -0754 00 4 00000          PXD 0,4
03626  0074 00 4 03577          TSX PRIN0,4
03627 -0534 00 4 03657  A4      LXD PS2,4
03630 -3 00000 4 03640          TXL     A6,4,0           EXIT IF NULL
03631  0500 00 4 00000          CLA     0,4              TEST FOR ATOM
03632  0734 00 4 00000          PAX     0,4
03633 -3 77776 4 03644          TXL     A2,4,-2          GO TO A2 IF NOT AN ATOM
03634  0500 00 0 03654          CLA     DOT              OTHERWISE PRINT IN DOT NOTATION
03635  0074 00 4 04076          TSX     $PRIN2,4         PUT IN PRINT LINE
03636  0500 00 0 03657          CLA     PS2              CDR OF LIST
03637  0074 00 4 03665          TSX     $PRIN1,4         PRINT AS ATOM
03640  0074 00 4 01736  A6      TSX     UNSAVE,4
03641  0500 00 0 03660          CLA RPAR2
03642 -0534 00 4 03656          LXD PS1,4
```

```
03643  0020 00 0 04076          TRA $PRIN2
03644  0500 00 0 03662   A2     CLA COMM2
03645  0074 00 4 04076          TSX $PRIN2,4
03646 -0534 00 4 03657          LXD PS2,4
03647  0500 00 4 00000          CLA 0,4
03650  0020 00 0 03622          TRA A3

03651  0500 00 0 03655   PRP2   CLA PRBLW
03652  0074 00 4 04076          TSX $PRIN2,4
03653  0020 00 0 03627          TRA A4
03654 -203360777777      OUT    OCT     603360777777         .
03655 -053143777777      PRBLW  OCT     453143777777         NIL
03656  0 00000 0 11177   PS1            $F4
03657  0 00000 0 00000   PS2
03660 +347777777777      RPAR2  OCT 347777777777
03661 -347777777777      LPAR2  OCT 747777777777
03662 -207777777777      COMM2  OCT 607777777777             BLANK    INSTEAD OF A COMMA
03663  0 00000 0 00000   CWRL
03664  0 00000 0 00000   L1

                         T      HED

                                      SUBROUTINE(PRIN1(L))
                                /     CAR(L)   N=-1 YIELDS ERROR
                                      ST = L
                                A1    CDR(L) = 0 YIELDS ERROR
                                      L = CDR(L)
                                      CAR(L) = PNAME YIELDS GO(A3)
                                      CAR(L) N= FLOAT YIELDS GO(A1)
                                      L = CAR(CDR(L))
                                      VAL = FLONAM(L)
                                      REPLACD(CONS(PNAME,CONS(VAL,CDR(ST))),ST)
                                      L = CDR(ST)
                                A3    L= CAR(CDR(L))
                                A2    PRIN2(CWR(CAR(L))
                                      L = CDR(L)
                                                     L=0 YIELDS RETURN
                                */    GO(A2)

03665 -0634 00 4 04060   PRIN1  SXD PR1,4
03666  0601 00 0 04057          STO PRSS                     SAVE OBJECT
                         ****CHANGE***********
03667 -0534 00 4 04273          LXD     WORDS,4
03670  3 00002 4 03672          TXH     *+2,4,2              IF NEAR END OF LINE
03671  0074 00 4 04230          TSX     TERPRI,4             THEN DO A TERPRI
03672 -0534 00 4 04057          LXD     PRSS,4
03673  0500 00 4 00000          CLA ,4
03674  0625 00 0 04071          STT     PTTGR
03675 -0320 00 0 00132          ANA ADDM
03676  0402 00 0 00132          SUB ADDM
03677  0100 00 0 03705          TZE PR3                      CAR(L) N=-1 YIELDS ERROR
03700 -0634 00 4 01530          SXD $ERROR,4
03701  0074 00 4 04230          TSX TERPRI,4
```

49299

```
03702 -0754 00 0 00000          PXD     0,0
03703  0074 00 4 01531          TSX $ERROR+1,4
03704  470145216344             BCI     1,PINATM        TRIED TO PRINT NON-OBJECT -PRIN1-
                    00132  ADDM SYN     $AMASK
03705  0520 00 0 04071  PR3     ZET     PTTGR
03706  0020 00 0 03720          TRA     PR3N
03707  0500 00 4 00000          CLA     0,4             FIRST WORD OF ATOM
03710  0020 00 0 03713          TRA     *+3
03711 -3 11226 4 03713  PR3P    TXL     *+2,4,$PNAME-1
03712 -3 11227 4 03735          TXL     PA3,4,$PNAME
03713 -0734 00 4 00000          PDX     0,4             CDR
03714 -3 00000 4 04045          TXL     PR5,4,0         UNPRINTABLE
03715  0500 00 4 00000          CLA     0,4             NEXT WORD
03716  0734 00 4 00000          PAX     0,4
03717  0020 00 0 03711          TRA     PR3P            EXAMINE WORD
03720 -0534 00 4 04057  PR3N    LXD     PR5S,4
03721  0500 00 4 00000          CLA     0,4
03722 -0734 00 4 00000          PDX     0,4
03723  0634 00 4 04070          SXA     PTPNT,4
03724  0500 00 0 04071          CLA     PTTGR
03725 -0320 00 0 00123          ANA     $QT2
03726 -0100 00 0 03751          TNZ     PR4F
03727  0500 00 0 04071          CLA     PTTGR
03730 -0320 00 0 00124          ANA     $QT4
03731 -0100 00 0 03754          TNZ     LUCY
03732 -0754 00 4 00000          PXD     0,4
03733  0074 00 4 03102          TSX     NUMNAM,4
03734  0020 00 0 03747          TRA     PR4E
                        *
03735 -0734 00 4 00000  PA3     PDX     0,4             FOUND A PNAME
03736  0500 00 4 00000          CLA     0,4
03737  0734 00 4 00000          PAX     0,4             POINTER TO PRINT LIST
03740  0500 00 4 00000  PR4     CLA     0,4             LIST ITEM
03741  0622 00 0 04061          STD     L               SAVE REST OF LIST IF ANY
03742  0734 00 4 00000          PAX     0,4             POINTER TO FIRST FULL FULL WORD
03743  0500 00 4 00000          CLA     0,4             FULL WORD
03744  0074 00 4 04076          TSX     $PRIN2,4        PRINT IT
03745 -0534 00 4 04061          LXD     L,4             PICK UP REST OF LIST
03746  3 00000 4 03740          TXH     PR4,4,0         PRINT MORE IF MORE
03747 -0534 00 4 04060  PR4E    LXD     PR1,4           EXIT BY RESTORING LINK IR
03750  0020 00 4 00001          TRA     1,4             EXIT
03751 -0754 00 4 00000  PR4F    PXD     0,4
03752  0074 00 4 04523          TSX     FLONAM,4
03753  0020 00 0 03747          TRA     PR4E
                        *
                        *       PRINT THE NUMBER OCTALLY
03754  0634 00 2 04043  LUCY    SXA     JUDY,2
03755  0534 00 2 04070          LXA     PTPNT,2         GET POINTER TO NUMBER
03756  0560 00 2 00000          LDQ     0,2
03757  0162 00 0 03764          TQP     BETTY           TEST FOR NEGATIVE NUMBER
03760  0500 00 0 04066          CLA     MISGN           IF SO, PRINT -
03761  0074 00 4 04076          TSX     $PRIN2,4
03762  0500 00 2 00000          CLA     0,2             REMOVE MINUS SIGN
03763 -0130 00 0 00000          XCL
03764 -0520 00 2 00000  BETTY   NZT     0,2             TEST IF NUMBER ALL ZEROS
```

```
03765  0020 00 0 04041          TRA     MARIE
                        *       LOOK FOR NON-ZERO DIGIT ON LEFT
03766 -0754 00 0 00000          PXD     ,0
03767  0774 00 2 00014          AXT     12,2                IR2 COUNTS ZEROS ON RIGHT
03770 -0763 00 0 00003          LGL     3
03771  1 77777 2 03772          TXI     *+1,2,-1            COUNT VACATED POSITIONS
03772  0100 00 0 03770          TZE     *-2
                        *       A NON-ZERO DIGIT HAS APPEARED ON THE LEFT
03773 -0501 00 0 15272          ORA     =64                 PUT IN OVERFLOW FLIPPER
03774  0140 00 0 03775          TOV     *+1                 SHUT OFF OVERFLOW LIGHT
03775 -0600 00 0 04062  GRETA   STQ     TONI                TEST IF ALL DIGITS ARE SPREAD
03776  0162 00 0 04000          TQP     *+2                 TEST FOR NON-ZERO SIGN BIT
03777  1 77777 2 04003          TXI     FIFI,2,-1           SOME DIGITS NOT SPREAD, SO CONTINUE
04000 -0520 00 0 04062          NZT     TONI
04001  0020 00 0 04020          TRA     DEBBY               TRA IF ALL NON-ZERO DIGITS SPREAD
04002  1 77777 2 04003          TXI     *+1,2,-1
04003  0767 00 0 00003  FIFI    ALS     3                   SPREAD ONE DIGIT
04004 -0763 00 0 00003          LGL     3
04005 -0140 00 0 03775          TNO     GRETA               SEE IF FULL WORD OF DIGITS
04006 -0600 00 0 04062          STQ     TONI                PRINT THE WORD
04007  0074 00 4 04076          TSX     $PRIN2,4
04010  0500 00 0 15245          CLA     =1                  PUT IN OVERFLOW FLIPPER
04011  0560 00 0 04062          LDQ     TONI
04012  0140 00 0 04013          TOV     *+1                 SHUT OFF OVERFLOW LIGHT
04013  0162 00 0 04015          TQP     *+2                 TEST FOR NON-ZERO SIGN BIT
04014  1 77777 2 04003          TXI     FIFI,2,-1
04015  0520 00 0 04062          ZET     TONI                SEE IF ALL DIGITS SPREAD
04016  1 77777 2 04003          TXI     FIFI,2,-1
04017  0020 00 0 04024          TRA     VICKI
                        *       FORM WORD FOR PRINTING
04020  0560 00 0 00140  DEBBY   LDQ     SEVENS              PUT 77S IN RIGHT END OF WORD
04021 -0763 00 0 00006          LGL     6                     OVERFLOW SIGNALS LEFT END OF WORD
04022 -0140 00 0 04021          TNO     *-1
04023  0074 00 4 04076          TSX     $PRIN2,4
                        *       PRINT Q AND SCALE FACTOR IF ANY
04024  3 00000 2 04027  VICKI   TXH     MICKY,2,0           CONTINUE IF 0 SCALE FACTOR
04025  0500 00 0 04067          CLA     BCIQ
04026  0020 00 0 04042          TRA     PATSY
04027 -3 00011 2 04034  MICKY   TXL     SANDY,2,9           TRA IF SCALE FACTOR LESS THAN 10
                        *       OCTAL SCALE FACTOR MORE THAN 10
04030 -0754 00 2 00000          PXD     ,2
04031  0400 00 0 04063          ADD     BQ10                FORM SCALE FACTOR FOR PRINTING
04032 -0760 00 0 00003          SSM
04033  0020 00 0 04042          TRA     PATSY
                        *       OCTAL SCALE FACTOR LESS THAN 10
04034 -0754 00 2 00000  SANDY   PXD     ,2
04035  0767 00 0 00006          ALS     6
04036  0400 00 0 04064          ADD     BQ0
04037 -0760 00 0 00003          SSM
04040  0020 00 0 04042          TRA     PATSY
04041  0500 00 0 04065  MARIE   CLA     BCIOQ               PRINT OQ
04042  0074 00 4 04076  PATSY   TSX     $PRIN2,4
04043  0774 00 2 00000  JUDY    AXT     **,2                ONLY THIS SECTION USES IX2
04044  0020 00 0 03747          TRA     PR4E
```

```
                        GENERATE A PRINT NAME FOR AN OBJECT WITHOUT ONE.

                        THE PRINT NAME IS OF THE FORM LDDDDC WHERE THE D,S ARE THE
                        OCTAL DIGITS OF THE 2,S COMPLIMENT OF THE FIRST WORD OF
                         THE PROPERTY LIST OF THE OBJECT.

04045 -0535 00 4 04057  PR5     LDC     PRSS,4
04046 -0754 00 4 00000          PXD     0,4
04047  0131 00 0 00000          XCA
04050  0074 00 4 10033          TSX     UCTALP,4
04051 -0501 00 0 15301          ORA     =HL00000
04052 -0760 00 0 00003          SSM                     FIX SIGN TO AGREE WITH P BIT FOR PRIN2
04053 -0760 00 0 00001          PBT
04054  0760 00 0 00002          CHS
04055 -0534 00 4 04060          LXD     PR1,4           RESTORE LINK IR
04056  0020 00 0 04076          TRA     $PRIN2          PUT IN PRINT LINE AND EXIT

04057  0 00000 0 00000  PRSS                            STORAGE FOR POINTER TO OBJECT
04060  0 00000 0 00000   PR1
04061  0 00000 0 00000   L
04062                    TONI   BSS     1
04063 +100066777777      BQ10   OCT     100066777777   USED TO FORM BCI QIN
04064 +100077777777      BQ0    OCT     100077777777   USED TO FORM BCI QN
04065 +005077777777      BCIOQ  OCT     005077777777   BCI OQ
04066 -007777777777      MISGN  OCT     407777777777   BCI -
04067 -107777777777      BCIQ   OCT     507777777777
04070                    PTPNT  BSS     1
04071  0 00000 0 00000  PTTGR                           TEST CELL FOR NUMBER FLAGS


                        PRIN2    PRINTS UP TO 6 CHARACTERS IN ONE WORD WHEN THE
                        CHARACTERS ARE JUSTIFIED TO THE LEFT AND FOLLOWED BY THE
                        ILLEGAL CHARACTER WHOSE OCTAL FORM IS 77

04072 -0634 00 4 04272  PRINT2 SXD PR9,4
04073 -0734 00 4 00000         PDX 0,4                  BRING BCD WORD TO AC
04074  0500 00 4 00000         CLA 0,4
04075  0020 00 0 04102         TRA     *+5
04076  3 00000 0 04344  PRIN2 TXH     $PUN2,,0         SWITCH TO PUNCH OUT ROUTINE
04077  0520 00 0 01522         ZET     $XPLSW
04100  0020 00 0 01507         TRA     $XPLG
04101 -0634 00 4 04272         SXD     PR9,4
04102 -0634 00 2 04271         SXD PR8,2
04103 -0634 00 1 04270         SXD PR7,1
04104 -0534 00 4 04273         LXD WORDS,4              ROOM LEFT IN OUTPUT RECORD
04105 -3 00000 4 04262         TXL INIT,4,0             CAN BE ZERO ONLY IF ROUTINE NOTUSED
04106  0774 00 1 00001  COMB4  AXT 1,1
04107  0601 00 0 04276         STO TEMP
04110 -0500 00 0 04276         CAL TEMP
04111 -0340 00 0 00140         LAS SEVENS               WORD OF ALL 77-S CAUSES NO ACTION
04112  0020 00 0 04114         TRA *+2
04113  0020 00 0 04145         TRA NOJOB
04114 -0320 00 0 15271  SHIFL  ANA =077     IS THE RIGHT CHARACTER 77
04115  0402 00 0 15271         SUB =077
```

```
04116 -0100 00 0 04123          TNZ JUST                    NOT 77
04117 -0500 00 0 04276          CAL TEMP
04120  0771 00 0 00006          ARS 6
04121  0602 00 0 04276          SLW TEMP
04122  1 00001 1 04114          TXI SHIFL,1,1
04123 -0500 00 0 04276  JUST    CAL TEMP
04124  0020 00 1 04133          TRA LSHIF+1,1
04125  0767 00 0 00006          ALS 6
04126  0767 00 0 00006          ALS 6
04127  0767 00 0 00006          ALS 6
04130  0767 00 0 00006          ALS 6
04131  0767 00 0 00006          ALS 6
04132  0602 00 0 04276  LSHIF   SLW TEMP
04133  0560 00 0 04276          LDQ TEMP
04134 -0500 00 0 04275          CAL PART
04135 -0534 00 2 04274          LXD PARTS,2
04136 -0763 00 0 00006  COMB    LGL 6
04137  0602 00 0 04275          SLW PART
04140 -2 00001 2 04152          TNX WFULL,2,1
04141  1 00001 1 04142  COMB5   TXI *+1,1,1
04142 -3 00006 1 04136          TXL COMB,1,6
04143 -0634 00 2 04274          SXD PARTS,2
04144 -0634 00 4 04273          SXD WORDS,4
04145 -0534 00 1 04270  NOJOB   LXD PR7,1
04146 -0534 00 2 04271          LXD PR8,2
04147 -0534 00 4 04272          LXD PR9,4
04150 -0754 00 0 00000          PXD 0,0
04151  0020 00 4 00001          TRA 1,4
04152  0602 00 4 04313  WFULL   SLW REC,4
04153 -2 00001 4 04156          TNX RECFL,4,1
04154  0774 00 2 00006          AXT 6,2
04155  0020 00 0 04141          TRA COMB5
04156 -0600 00 0 04276  RECFL   STQ TEMP
04157                           TSSX    WRFLX,4
04160  0 00014 0 04277                  REC-12,,12     ***LENGTH OF OUTPUT LINE DECREASED***
04161  0560 00 0 04276          LDQ TEMP
04162  0774 00 4 00014          AXT     12,4
04163 -0500 00 0 15311          CAL     =H
04164  0602 00 0 04275          SLW     PART
04165  0774 00 2 00005          AXT     5,2
04166  0020 00 0 04141          TRA     COMB5
                        *
                        *       TAB     TAKES NUMBER FROM 1 TO 20 AND TABS OUTPUT LINE
                        *               OUT THAT MANY WORDS.
                        *
04167  0634 00 4 04222   TAB    SXA     TABX,4
                       04167    TABTAB SYN     TAB                 GET AROUND HEADING NONSCENCE
04170  0634 00 2 04223          SXA     TABY,2
04171 -0734 00 4 00000          PDX     0,4
04172  0500 00 4 00000          CLA     0,4
04173 -0734 00 4 00000          PDX     0,4
04174  0500 00 4 00000          CLA     0,4
04175  0734 00 4 00000          PAX     0,4
04176 -3 00000 4 04222          TXL     TABX,4,0        NO TAB AT ALL
04177  3 00023 4 04226          TXH     TABC,4,19
```

```
04200 -0634 00 4 04202          SXD     *+2,4
04201  0774 00 4 00024          AXT     20,4
04202  2 00000 4 04203          TIX     *+1,4,**
04203 -0634 00 4 04220          SXD     TABA,4          TAB OUT TO HERE
04204 -0534 00 4 04273          LXD     WORDS,4
04205 -3 00001 4 04226          TXL     TABC,4,1
04206 -0534 00 2 04274          LXD     PARTS,2
04207  0560 00 0 15311          LDQ     =H
04210  0500 00 0 04275          CLA     PART
04211 -0763 00 0 00006          LGL     6
04212  2 00001 2 04211          TIX     *-1,2,1
04213  0774 00 2 00006          AXT     6,2
04214 -0634 00 2 04274          SXD     PARTS,2
04215  0602 00 4 04313  TABB    SLW     REC,4
04216 -0500 00 0 15311          CAL     =H
04217  1 77777 4 04220          TXI     *+1,4,-1
04220  3 00000 4 04215  TABA    TXH     TABB,4,**
04221 -0634 00 4 04273          SXD     WORDS,4
04222  0774 00 4 00000  TABX    AXT     **,4
04223  0774 00 2 00000  TABY    AXT     **,2
04224 -0754 00 0 00000          PXD     0,0
04225  0020 00 4 00001          TRA     1,4
                       *
04226  0074 00 4 04230  TABC    TSX     TERPRI,4        DUMP THIS LINE
04227  0020 00 0 04222          TRA     TABX
                       *
                       *        TERPRI  FINISHES LINE OUT WITH BLANKS AND PRINTS IT
                       *
04230 -0634 00 2 04271  TERPRI  SXD PR8,2
04231 -0634 00 4 04272          SXD PR9,4
04232 -0534 00 2 04274          LXD PARTS,2
04233 -0534 00 4 04273          LXD WORDS,4
04234 -0500 00 0 04275          CAL PART
04235  0560 00 0 15311          LDQ =H
04236 -0763 00 0 00006  TER1    LGL 6
04237  2 00001 2 04236          TIX TER1,2,1
04240  0602 00 4 04313  TER3    SLW REC,4
04241 -2 00001 4 04244          TNX TER2,4,1
04242 -0500 00 0 15311          CAL =H
04243  0020 00 0 04240          TRA TER3
04244  0520 00 0 01522  TER2    ZET     $XPLSW
04245  0020 00 0 04250          TRA     *+3
04246                           TSSX    WRFLX,4
04247  0 00014 0 04277                  REC-12,,12    LENGTH OF OUTPUT HAS BEEN CHANGED****
04250  0774 00 4 00014          AXT     12,4
04251 -0634 00 4 04273          SXD WORDS,4
04252  0774 00 2 00005          AXT     5,2
04253 -0634 00 2 04274          SXD PARTS,2
04254 -0534 00 2 04271          LXD PR8,2
04255 -0534 00 4 04272          LXD PR9,4
04256  0500 00 0 15311          CLA =H
04257  0601 00 0 04275          STO PART
04260 -0754 00 0 00000          PXD 0,0
04261  0020 00 4 00001          TRA 1,4
04262  0774 00 4 00014  INIT    AXT     12,4
```

```
04263  0560 00 0 15311          LDQ     =H
04264 -0600 00 0 04275          STQ PART
04265  0774 00 2 00005          AXT 5,2
04266 -0634 00 2 04274          SXD PARTS,2
04267  0020 00 0 04106          TRA COMB4
                         *
04270  0 00000 0 00000  PR7
04271  0 00000 0 00000  PR8
04272  0 00000 0 00000  PR9
04273  0 00014 0 00000 WORDS           ,,12
04274  0 00005 0 00000 PARTS           ,,5
04275  0 00000 0 00000  PART
04276  0 00000 0 00000  TEMP
04313                   REC     BES     12
04313  0 00000 0 00000  PCPPI                           PUSHDOWN COUNTER STORAGE
                         *
                         * BCDAD1        A CONVERT TABLE FOR ADDING 1 TO A 6 DIGIT BCD NUMBER
                         *               USED BY LOADING BCD NUMBER INTO AC AND DOING
                         *       CVR     BCDAD1,,6
                         *
04314  0 00000 0 04314  ADT     PZE     ADT                     0
04315  0 10000 0 04314 BCDAD1  PZE     ADT,,1*4096             1
04316  0 20000 0 04314          PZE     ADT,,2*4096
04317  0 30000 0 04314          PZE     ADT,,3*4096             3
04320  0 40000 0 04314          PZE     ADT,,4*4096             4
04321  0 50000 0 04314          PZE     ADT,,5*4096             5
04322  0 60000 0 04314          PZE     ADT,,6*4096             6
04323  0 70000 0 04314          PZE     ADT,,7*4096             7
04324  1 00000 0 04314          PON     ADT                     8
04325  1 10000 0 04314          PON     ADT,,1*4096             9
04326  0 00000 0 04315          PZE     BCDAD1                  10
```

LISP FUNCTIONS

```
                                        INSERT  $PUNCHF
                                * PUNCH         WRITES OUT A LIST ON THE SYSTEM PERIFIAL PUNCH TAPE
                                *               (SYSPPT) IN A FORM SUTABLE FOR PUNCHING IN BCD.
                                *
 04327  0634 00 4 04342   PUNCH SXA     PNCHX,4             SAVE LINK IR
 04330  0600 00 0 04522         STZ     PUNLM          ***INDICATE FIRST 72 COLUMNS***
 04331 -0625 00 0 04475         STL     PUNACT              ACTIVTE PUNCH ROUTINE
 04332 -0734 00 4 00000         PDX     0,4                 ARGUMENT TO IR 4
 04333  0502 00 0 04076         CLS     $PRIN2              SET SWITCH TO
 04334  0601 00 0 04076         STO     $PRIN2              GO TO PUNCH ROUTINE
 04335 -0754 00 4 00000         PXD     0,4                 ARGUMENT TO AC
 04336  0601 00 0 02642         STO     PRINTL         SAVE THE ARGUMENT
 04337  0074 00 4 03577         TSX     $PRINO,4            USES PRINT ROUTINE
 04340  0074 00 4 04434         TSX     TERPUN,4            TERMINATE PUNCHING
 04341  0500 00 0 02642         CLA     PRINTL         RESTORE THE ARGUMENT
 04342  0774 00 4 00000   PNCHX AXT     **,4                RESTORE LINK IR
 04343  0020 00 4 00001         TRA     1,4                 EXIT
                                *
                                * PUN2          PUNCH EQUIVELENT OF PRIN 2
                                *
 04344  0634 00 4 04377   PUN2  SXA     PNX,4               SAVE INDEX REGISTERS
 04345  0634 00 2 04400         SXA     PNY,2
 04346  0634 00 1 04401         SXA     PNZ,1
 04347  0774 00 4 00022   PWRDS AXT     18,4                NUMBER OF WORDS LEFT IN BUFFER
 04350  0774 00 2 00006   PPRTS AXT     6,2                 CHARACTER POSITION
 04351  0774 00 1 00006         AXT     6,1                 MAXIMUM NUMBER OF CHARACTERS
 04352  0131 00 0 00000         XCA                         ARGUMENT TO MQ
 04353 -0754 00 0 00000   PLP   PXD     0,0                 CLEAR AC
 04354 -0763 00 0 00006         LGL     6                   CHARACTER TO MQ
 04355  0520 00 0 04522         ZET     PUNLM          FIRST 72 CHARACTERS
 04356  0340 00 0 15270         CAS     =HC0000        NO, LOOK FOR BLANK
 04357  0020 00 0 04365         TRA     PUND           NOT FOUND,CONTINUE LINE
 04360  0020 00 0 04362         TRA     *+2            FOUND
 04361  0020 00 0 04365         TRA     PUND           NOT FOUND
 04362 -0600 00 0 04474         STQ     PNCQ           SAVE REST OF MQ
 04363 -3 00005 2 04447         TXL     TPLP-1,2,5
 04364  1 00001 4 04455         TXI     PUNWD,4,1      WORD EMPTY,DO NOT PUNCH IT
 04365  0340 00 0 15271   PUND  CAS     =077                COMPARE WITH 77
 04366  0761 00 0 00000         NOP                         GREATER, (IMPOSSIBLE)
 04367  0020 00 0 04374         TRA     PUUT                = , GO TO EXIT
 04370  0522 00 2 04473         XEC     PCNT,2              LESS THAN, SHIFT CHARACTER
 04371 -0602 00 4 04521         ORS     POUP,4              PUT IN OUTPUT LINE
 04372 -2 00001 2 04403         TNX     PRPLP,2,1           GO IF LAST CHARACTER IN WORD
 04373  2 00001 1 04353   PGRA  TIX     PLP,1,1             GET NEXT CHARACTER
 04374  0634 00 2 04350   PUUT  SXA     PPRTS,2             SAVE INDEX 2 N 4
 04375  0634 00 4 04347         SXA     PWRDS,4
 04376 -0754 00 0 00000         PXD     0,0                 CLEAR AC
 04377  0774 00 4 00000   PNX   AXT     **,4                RESTORE INDEX REGISTERS
 04400  0774 00 2 00000   PNY   AXT     **,2
 04401  0774 00 1 00000   PNZ   AXT     **,1
 04402  0020 00 4 00001         TRA     1,4                 EXIT
                                *
 04403  0774 00 2 00006   PRPLP AXT     6,2                 RELOAD CHARACTER COUNT
 04404  1 77777 4 04405         TXI     *+1,4,-1
 04405  3 00007 4 04373         TXH     PGRA,4,7       FIRST 11 WORDS,NO PROBLEM
```

```
04406 -0625 00 0 04522          STL     PUNLM           INDICATE BREAK ON FIRST CPPORTUNITY
04407  3 00000 4 04373          TXH     PGRA,4,0        LINE NOT FULL YET
04410 -0600 00 0 04474          STQ     PNCQ             YES,IT IS PUNCH AWAY
04411  0020 00 0 04455          TRA     PUNWD            WITHOUT REGARD TO NICETIES
04412  0520 00 0 01056  PUNWRT ZET     $OUTSW
04413  0020 00 0 04416          TRA     WRITIT
04414 -0625 00 0 01056          STL     $OUTSW
04415  0074 00 4 01106          TSX     $ASIGN,4
04416                   WRITIT CALLIO   WRWAIT,OUTFIL,(=0,(OUTBUF-1,,**),DISKER)  FILL IN AT W+5
                        *END OF CHANGES
04425  0774 00 4 00022          AXT     18,4            MAX NO. OF WCRDS*****
04426  0600 00 4 04521          STZ     POUP,4               ZERO OUTPUT BUFFER
04427  2 00001 4 04426          TIX     *-1,4,1
04430  0774 00 4 00022          AXT     18,4                 RELOAD WORD COUNT
04431  0600 00 0 04522          STZ     PUNLM           RESET COL. 72 SWITCH****
04432  0560 00 0 04474          LDQ     PNCQ                 RESTORE CONTENTS OF MQ
04433  0020 00 0 04373          TRA     PGRA                 CONTINUE WORK
                        *
                        * TERPUN        FILLS OUT BUFFER WITH BLANKS AND PUNCHES OUT LAST CARD
                        *               OPERATES ONLY IF PUNCH ROUTINE IS CURRENTLY ACTIVE
                        *
04434 -0520 00 0 04475  TERPUN NZT      PUNACT               SKIP IF PUNCH ROUTINE IS CURRENTLY ACT
04435  0020 00 4 00001          TRA     1,4                  IMMEDIATE EXIT
04436  0600 00 0 04475          STZ     PUNACT               DE ACTIVATE THE PUNCH ROUTINE
04437  0634 00 4 04377          SXA     PNX,4                SAVE INDEX REGISTERS
04440  0634 00 2 04400          SXA     PNY,2
04441  0634 00 1 04401          SXA     PNZ,1
04442  0500 00 0 04076          CLA     $PRIN2
04443  0602 00 0 04076          SLW     $PRIN2               RESTORE PRIN2 SWITCH
04444  0534 00 4 04347          LXA     PWRDS,4              PICK UP WORD COUNT
04445  0534 00 2 04350          LXA     PPRTS,2              CHARACTER COUNT
04446  0774 00 1 00001          AXT     1,1                  CONSTANT 1
04447  0560 00 0 04521          LDQ     S57             LOAD WORD OF 57Q
04450 -0754 00 0 00000  TPLP    PXD     0,0                  CLEAR AC
04451 -0763 00 0 00006          LGL     6                    1 INTO AC
04452  0522 00 2 04473          XEC     PCNT,2               SHIFT INTO POSITIN
04453 -0602 00 4 04521          ORS     POUP,4               PUT IN OUTPUT LINE
04454  2 00001 2 04450          TIX     TPLP,2,1             FILL OUT THIS WORD
04455  0756 00 4 00000  PUNWD   PCA     0,4             NO. OF WORDS LEFT IN LINE
04456  0400 00 0 15261          ADD     =19             YIELDS NO. OF WORDS WRITTEN
04457  0621 00 0 04476          STA     OUTBUF-1        SETUP OUTPUT LINEMARK
04460  0400 00 0 15245          ADD     =1
04461  0767 00 0 00022          ALS     18
04462  0622 00 0 04423          STD     WRITIT+5        SETUP WRITE-WAIT COMMAND
04463  0774 00 2 00006          AXT     6,2
04464  0020 00 0 04412          TRA     PUNWRT          WRITE OUT LINE
                                *        COSTANTS, STORAGE AND SHIFT TABLE
04465  0767 00 0 00036          ALS     30
04466  0767 00 0 00030          ALS     24
04467  0767 00 0 00022          ALS     18
04470  0767 00 0 00014          ALS     12
04471  0767 00 0 00006          ALS     6
04472  0761 00 0 00000          NOP
04473  0 00000 0 00000  PCNT    PZE                          BASE OF SHIFT TABLE AND CARD COUNT
04474  0 00000 0 00000  PNCQ
```

LISP FUNCTIONS

```
04475   0 00000 0 00000  PUNACT                          NON-ZERO IF PUNCH ROUTINE ACTIVE
04476  -377777000000              OCT     777777000000  LINE MARK
04477  +000000000000     OUTBUF   OCT     0,0,0,0,0,0   THERE SHOULD BE 18 OF THESE
04505  +000000000000              OCT     0,0,0,0,0,0   ..
04513  +000000000000              OCT     0,0,0,0,0,0   ..
04521  -175757575757     S57      OCT     575757575757
                   04521 POUP     SYN     S57
04522   0 00000 0 00000  PUNLM    PZE                   ARE WE WORKING ON IST 72 CHARACTERS
```

LISP FUNCTIONS

```
                                   INSERT  $FLONMF

                                       FLONAM                    MAY 14,1559
                                         FORMS THE  BCD LIST  FOR A FLOATING NUMBER IN THE ACC

                           T       HED
04523  0634 00 4 04711  FLONAM SXA     FLNX,4
04524 -0734 00 4 00000         PDX     0,4
04525  0500 00 4 00000         CLA     0,4
04526 -0100 00 0 04535         TNZ     FLNA
04527  0534 00 4 04711         LXA     FLNX,4
04530  0131 00 0 00000         XCA
04531  0500 00 0 04731         CLA     FLZPZ              0.0
04532  0162 00 0 04076         TQP     $PRIN2
04533  0402 00 0 04754         SUB     CO                 -0.0
04534  0020 00 0 04076         TRA     $PRIN2
04535  0634 00 2 04712  FLNA   SXA     FLNY,2
04536  0634 00 1 04713         SXA     FLNZ,1
04537  0774 00 1 00001         AXT     1,1                SET UP BUFFER IRS
04540  0774 00 2 00044         AXT     36,2
04541  0600 00 0 04726         STZ     FLOPB-3
04542  0600 00 0 04727         STZ     FLOPB-2
04543  0600 00 0 04730         STZ     FLOPB-1
04544  0601 00 0 00051         STO     COMMON+5
04545  0131 00 0 00000         XCA
04546 -0754 00 0 00000         PXD     ,0                CLEAR ACC. AND SIGN.          21303
04547  0765 00 0 00000         LRS     0                 SIGN TO MQ                    21305
04550  0763 00 0 00010         LLS     8                 CHARACTERISTIC.               21310
04551  0402 00 0 15273         SUB     =128              128                           21315
04552 -0600 00 0 00044         STQ     COMMON            SAVE MANTISSA.                21320
04553  0131 00 0 00000         XCA                       MULTIPLY BY                   21325
04554  0200 00 0 04743         MPY     LOG2              LOG BASE 10 OF 2.             21330
04555  0601 00 0 00046         STO     COMMON+2                                        21335
04556  0120 00 0 04565         TPL     FL75                                            21340
04557  0402 00 0 04742         SUB     A1                1                             21345
04560  0601 00 0 00046         STO     COMMON+2                                        21350
04561  0131 00 0 00000         XCA                                                     21355
04562  0760 00 0 00006         COM                                                     21360
04563  0760 00 0 00003         SSP                                                     21365
04564  0131 00 0 00000         XCA                                                     21370
04565  0200 00 0 04744  FL75   MPY     LOG10             LOG BASE 2 OF 1074.           21375
04566  0765 00 0 00041         LRS     33                                              21380
04567  0621 00 0 04600         STA     FL76A                                           21385
04570 -0600 00 0 00054         STQ     COMMON+8                                        21390
04571  0774 00 4 00007         AXT     7,4                                             21395
04572  0560 00 0 04745         LDQ     C7                                              21400
04573  0200 00 0 00054  FL76   MPY     COMMON+8                                        21405
04574  0400 00 4 04755         ADD     CO+1,4                                          21410
04575  0131 00 0 00000         XCA                                                     21415
04576  2 00001 4 04573         TIX     FL76,4,1                                        21420
04577  0200 00 0 00044         MPY     COMMON            MANTISSA.                     21425
04600  0774 00 4 00000  FL76A  AXT     **,4                                            21427
04601  0765 00 4 00042         LRS     34,4                                            21430
04602  0100 00 0 04607         TZE     FL77                                            21435
04603  0221 00 0 04741         DVP     A1-1              10.                           21440
```

```
04604  0500 00 0 00046          CLA    COMMON+2                                21445
04605  0400 00 0 04742          ADD    A1              1.                      21450
04606  0601 00 0 00046          STO    COMMON+2                                21455
04607  0774 00 4 00010   FL77   AXT    8,4
04610 -0200 00 4 04742          MPR    A1,4            TO TO DEC. PLACES.      21475
04611  0340 00 4 04742          CAS    A1,4                                    21480
04612  0761 00 0 00000          NOP                    GREATER.                21485
04613  0020 00 0 04615          TRA    FL79            EQUAL.                  21490
04614  0020 00 0 04621          TRA    FL80            LESS.                   21495
04615  0500 00 0 04742   FL79   CLA    A1              ROUNDING CAUSED CARRY.  21500
04616  0400 00 0 00046          ADD    COMMON+2                                21505
04617  0601 00 0 00046          STO    COMMON+2        EXP+1.                  21510
04620  0500 00 4 04743          CLA    A1+1,4          TO TO THE DEC. PL.-1.   21515
04621  0601 00 0 00054   FL80   STO    COMMON+8                                21520
04622 -0754 00 0 00000          PXD    ,0                                      21525
04623  0560 00 0 00046          LDQ    COMMON+2        ENTER DEC EXP.          21530
04624  0221 00 0 04741          DVP    A1-1            10                      21535
04625 -0600 00 0 00053          STQ    COMMON+7                                21540
04626  0634 00 4 04644          SXA    FL82,4                                  21545
04627  0074 00 4 04716          TSX    INBCD,4                                 21550
04630 -0754 00 0 00000          PXD    ,0                                      21555
04631  0560 00 0 00053          LDQ    COMMON+7                                21560
04632  0221 00 0 04741          DVP    A1-1                                    21565
04633  0100 00 0 04635          TZE    *+2
04634  0074 00 4 04716          TSX    INBCD,4                                 21570
04635  0500 00 0 00046          CLA    COMMON+2                                21575
04636  0100 00 0 04642          TZE    FL81                                    21580
04637  0120 00 0 04642          TPL    FL81                                    21585
04640  0500 00 0 15264          CLA    =H00000-
04641  0074 00 4 04716          TSX    INBCD,4                                 21605
04642  0500 00 0 15262   FL81   CLA    =H00000E
04643  0074 00 4 04716          TSX    INBCD,4
04644  0774 00 4 00000   FL82   AXT    **,4                                    21620
04645  0600 00 0 04725          STZ    FLZET
04646  0500 00 0 00054   FL65   CLA    COMMON+8                                21145
04647  0765 00 0 00043          LRS    35
04650  0221 00 0 04741          DVP    A1-1            10.                     21165
04651 -0600 00 0 00054          STQ    COMMON+8        FRACTIONAL PART.        21170
04652 -0520 00 0 04725          NZT    FLZET
04653  0100 00 0 04660          TZE    FL01
04654 -0602 00 0 04725          ORS    FLZET
04655  0634 00 4 04657          SXA    *+2,4           SAVE IR4.               21175
04656  0074 00 4 04716          TSX    INBCD,4         ENTER DIGIT.            21180
04657  0774 00 4 00000          AXT    **,4            RESTORE.                21185
04660  2 00001 4 04646   FL01   TIX    FL65,4,1                                21190
04661 -0754 00 0 00000          PXD    0,0
04662 -0520 00 0 04725          NZT    FLZET
04663  0074 00 4 04716          TSX    INBCD,4
04664  0500 00 0 15263          CLA    =H00000.        DEC. POINT.             21195
04665  0074 00 4 04716          TSX    INBCD,4         ENTER.                  21200
04666 -0754 00 0 00000          PXD    0,0
04667  0074 00 4 04716          TSX    INBCD,4
04670  0560 00 0 00051          LDQ    COMMON+5
04671 -0500 00 0 15270          CAL    =H00000
04672  0162 00 0 04674          TQP    FL70            FOR PLUS.               21225
```

```
04673 -0500 00 0 15264          CAL     =H00000-
04674  0074 00 4 04716   FL70   TSX     INBCD,4     INSERT BLANK OR MINUS.          21235
04675 -0754 00 2 00000          PXD     0,2
04676 -0737 00 2 00000          PDC     0,2
04677  0560 00 0 00140          LDQ     ONES        FILL OUT LAST WORD WITH 77S
04700 -0500 00 1 04731          CAL     FLOPB,1
04701 -0763 00 2 00000          LGL     0,2
04702 -0130 00 0 00000          XCL
04703  0131 00 0 00000          XCA
04704  0074 00 4 04076          TSX     $PRIN2,4
04705 -2 00001 1 04711          TNX     FLNX,1,1
04706  0500 00 1 04731          CLA     FLOPB,1
04707  0074 00 4 04076          TSX     $PRIN2,4
04710  2 00001 1 04706          TIX     *-2,1,1
04711  0774 00 4 00000   FLNX   AXT     **,4
04712  0774 00 2 00000   FLNY   AXT     **,2
04713  0774 00 1 00000   FLNZ   AXT     **,1
04714 -0754 00 0 00000          PXD     0,0
04715  0020 00 4 00001          TRA     1,4
                       *
04716 -0320 00 0 15271   INBCD  ANA     =077
04717  0767 00 2 00044          ALS     36,2
04720 -0602 00 1 04731          ORS     FLOPB,1
04721  2 00006 2 04724          TIX     *+3,2,6
04722  1 00001 1 04723          TXI     *+1,1,1
04723  0774 00 2 00044          AXT     36,2
04724  0020 00 4 00001          TRA     1,4
                       *
04725  0 00000 0 00000   FLZET
04731                    FLOPB  BES     3
04731 -200033007777      FLZPZ  OCT     600033007777
04732 +000575360400             DEC     1E8B35,1E7B35,1E6B35,1E5B35,1E4B35,1E3B35,1E2B35,1E1B35
04742 +000000000001      A1     DEC     1                                           70108
04743 +115040465025      LOG2   OCT     115040465025      LOG BASE 10 OF 2.         70110
04744 +324464741127      LOG10  OCT     324464741127      LOG BASE 2 OF 10-4.       70115
04745 +000001601225      C7     OCT     1601225,7762664,132240566,1164125106,7066267024
04752 +036577252307             OCT     36577252307,130562064437
04754  2 00000 0 00000   C0     TIX     0,0,0                                       70155
                00140    ONES   SYN     SEVENS
```

LISP FUNCTIONS

```
                    INSERT  $REDZIF
                *
                    READ

                      READ = SELECT(RD.,LPAR,READ1.,
                                    LITER,INTERN.,
                                    NUMB,INTERN.,
                                    RPAR,ERROR.,
                                    I,ERROR)

                    READ1

                    READ1 = SELECT(RD.,RPAR,0.,
                                   LPAR,CONS(READ1,READ1).,
                                   LITER,CONS(INTERN,READ1).,
                                   NUMB,CONS(INTERN,READ1))

                I       HED
04755  0634 00 4 04757  READ  SXA     REDS1,4          SAVE LINK IR
04756  0074 00 4 05047        TSX     $RD,4            GET FIRST ITEM
04757  0774 00 4 00000  REDS1 AXT     **,4             RESTORE LINK IR
04760  0340 00 0 05043  REDIS CAS     RLPAR            DISPATCH ON TYPE OF ITEM READ
04761  0020 00 0 04763        TRA     *+2
04762  0020 00 0 05000        TRA     READ1            WAS (
04763  0340 00 0 05044        CAS     RRPAR
04764  0020 00 0 04766        TRA     *+2
04765  0020 00 0 04772        TRA     REDER
04766  0340 00 0 05045        CAS     RDOT
04767  0020 00 4 00001        TRA     1,4
04770  0020 00 0 04772        TRA     REDER
04771  0020 00 4 00001        TRA     1,4
04772 -0634 00 4 01530  REDER SXD     $ERROR,4         MUST BE AN ERROR
04773  0601 00 0 02641        STO     RS2              SAVE TYPE
                      *****THE PRINTING OF THE INPUT BUFFER HAS BEEN SUPPRESSED*******
04774  0074 00 4 05452        TSX     TEREAD,4      **TERMINATE READING****
04775  0500 00 0 02641        CLA     RS2              GET TYPE
04776  0074 00 4 01531        TSX     $ERROR+1,4       GO TO ERROR
04777  512521242551           BCI     1,READER         CONTEXT ERROR
                *
05000 -0634 00 4 02640  READ1 SXD     RS1,4            SAVE LINK IR
05001  0074 00 4 05047        TSX     $RD,4            GET NEXT ITEM
05002  0340 00 0 05044        CAS     RRPAR
05003  0020 00 0 05005        TRA     *+2
05004  0020 00 0 05024        TRA     RP1              WAS )  RETURN WITH NIL
05005  0074 00 4 01723        TSX     $SAVE,4
05006 -3 02643 0 01766        TXL     $END2,,RS2+2     SAVE 2 ITEMS
05007  0340 00 0 05045        CAS     RDOT
05010  0020 00 0 05012        TRA     *+2
05011  0020 00 0 05027        TRA     RP2              WAS .
05012  0340 00 0 05043        CAS     RLPAR
05013  0020 00 0 05015        TRA     *+2
05014  0074 00 4 05000        TSX     READ1,4
05015  0601 00 0 02641        STO     RS2              SAVE RESULTS
05016  0074 00 4 05000        TSX     READ1,4          GET NEXT ITEM
```

```
05017  0131 00 0 00000          XCA                    PUT IN MQ
05020  0500 00 0 02641          CLA     RS2            FIRST ITEM
05021  0074 00 4 01736          TSX     UNSAVE,4
05022 -0534 00 4 02640          LXD     RS1,4          RESTORE LINK IR
05023  0020 00 0 02714          TRA     $CONS          CONSTRUCT A LIST
                        *
05024 -0754 00 0 00000  RP1      PXD     0,0            WAS )  RETURN WITH NIL
05025 -0534 00 4 02640           LXD     RS1,4
05026  0020 00 4 00001           TRA     1,4
                        *
05027  0074 00 4 05047  RP2      TSX     $RD,4          WAS .  GET NEXT ITEM
05030  0074 00 4 04760           TSX     REDIS,4        DISPATCH ON IT
05031  0601 00 0 02641           STO     RS2            SAVE RESULTS
05032  0074 00 4 05047           TSX     $RD,4          GET NEXT ITEM
05033  0340 00 0 05044           CAS     RRPAR          SHOULD BE )
05034  0020 00 0 04772           TRA     REDER          GO TO ERROR IF NOT
05035  0020 00 0 05037           TRA     *+2
05036  0020 00 0 04772           TRA     REDER
05037  0500 00 0 02641           CLA     RS2            GET ITEM READ
05040  0074 00 4 01736           TSX     UNSAVE,4
05041 -0534 00 4 02640           LXD     RS1,4          RESTORE LINK IR
05042  0020 00 4 00001           TRA     1,4            RERURN WITH IT
                        *
                     00152 RLTR  SYN     QUOTED         SYMBOL FLAG
                        *
                        I        HED

                                 RD(A)


05043  0 00175 0 00000  RLPAR            ,,$H74D
05044  0 00173 0 00000  RRPAR            ,,$H34D
05045  0 00172 0 00000  RDOT             ,,$H33D
05046                    RDVAL BSS       0
05046  0 00000 0 00001  LRCIS            1              CARD IMAGE EMPTY TEST CELL
05047  0500 00 0 06402  RD       CLA     RDLST
05050  0100 00 0 05053           TZE     RDAA           GO IF NOT
05051  0600 00 0 06402           STZ     RDLST          OTHERWISE ZERO
05052  0020 00 4 00001           TRA     1,4            AND EXIT
05053  0634 00 4 05111  RDAA     SXA     RDX,4          SAVE INDEX REGISTERS
05054  0634 00 2 05116           SXA     RDY,2
05055  0634 00 1 05115           SXA     RDZ,1
05056  0604 00 0 06401           STI     RDIND          SAVE THE INDICATORS
05057  0441 00 0 15244           LDI     =0
05060  0500 00 0 15255           CLA     =12            PREPARE FOR TY IO***
05061  0520 00 0 05276           ZET     PRDIND         WHAT WAS PREVIOUS READ
05062  0020 00 0 05120           TRA     RDMOSE         IT CAME FROM DISK
05063  0520 00 0 01215           ZET     RDINDC         WHAT IS CURRENT READ
05064  0020 00 0 05124           TRA     RDMOS1         DISK IS CURRENT
05065  0774 00 2 00006  RDPTS    AXT     6,2                SET UP IR 2 AND 1
05066  0774 00 1 00660  RDWDS    AXT     432,1
05067  0074 00 4 05233  RDGC     TSX     GET,4          GET THE FIRST CHARACTER
05070  0734 00 4 00000           PAX     0,4            TYPE TO INDEX REGISTER
05071  0020 00 4 05101           TRA     RDJT1,4        DISPATCH ON TYPE
05072  0020 00 0 05133           TRA     RDDLR          $
```

```
05073  0020 00 0 05163          TRA    RDLT
05074  0020 00 0 05164          TRA    RDNM          NUMBER
05075  0020 00 0 05067          TRA    RDGC          ,
05076  0020 00 0 05110          TRA    RDPU          (
05077  0020 00 0 05110          TRA    RDPU          )
05100  0020 00 0 05110          TRA    RDPU          .
05101                    RDJT1 TSSX   WRFLX,4        ILLEGAL CHARACTER
05102  0 00016 0 05517           LWPO,,14
05103  0074 00 4 05452          TSX    TEREAD,4      ***TERMINATE READING****
05104 -0754 00 0 00000          PXD    0,0           CLEAR AC
05105                           ERROR  ILCHAR
05110  0500 00 4 05046  RDPU    CLA    RDVAL,4
05111  0774 00 4 00000  RDX     AXT    **,4
05112  0634 00 2 05065          SXA    RDPTS,2       SAVE INDEX REGISTERS
05113  0634 00 1 05066          SXA    RDWDS,1
05114  0441 00 0 06401          LDI    RDIND         RESTORE INDICATORS
05115  0774 00 1 00000  RDZ     AXT    **,1          RESTORE INDEX REGISTERS
05116  0774 00 2 00000  RDY     AXT    **,2
05117  0020 00 4 00001          TRA    1,4           EXIT
05120  0500 00 0 15274  RDMOSE  CLA    =432          PREVIOUS READ WAS FROM DISK
05121 -0520 00 0 01215          NZT    RDINDC        WHAT IS CURRENT DEVICE
05122  0020 00 0 05124          TRA    RDMOS1        TY IS CURRENT
05123  0020 00 0 05065          TRA    RDPTS         NO CHANGE IN DEVICE
                         ***NONZERO INDICATORS ARE FOR TY ZERO FOR DISK
05124  0600 00 0 06402  RDMOS1  STZ    RDLST         CHANGE IN DEVICE,INITIALIZE
05125 -0625 00 0 05046          STL    LRCIS         INDICATE NEW RECORD NEEDED
05126  0621 00 0 05066          STA    RDWDS         INITIALIZE IR1
05127  0500 00 0 15251          CLA    =6
05130  0621 00 0 05065          STA    RDPTS
05131  0621 00 0 05410          STA    PUTMC         INITIALIZE IR2 AND PUT
05132  0020 00 0 05065          TRA    RDPTS
                         *
05133  0055 00  000003 RDDLR   SIR    3             SET FIRST CHARCTER AND LITERAL INDICAT
05134  0074 00 4 05233          TSX    GET,4         IS NEXT CHARACTER A $
05135  0734 00 4 00000          PAX    0,4           IF SO IT INDICATES A LITERAL STRING
05136  0500 00 0 06403          CLA    GTVAL         SAVE VALUE OF GET
05137  0601 00 0 06400          STO    RDDDC
05140  3 00006 4 05151          TXH    RDDD,4,6      GO IF A $
05141  0634 00 4 05147          SXA    RDT,4         NOT SO DO A REGULAR D
05142  0500 00 0 15266          CLA    =H00000$
05143  0601 00 0 06403          STO    GTVAL
05144  0074 00 4 05403          TSX    PUT,4         PUT IN OUTPUT BUFFER
05145  0500 00 0 06400          CLA    RDDDC         LAST VALUE OF GET
05146  0601 00 0 06403          STO    GTVAL
05147  0774 00 4 00000  RDT     AXT    **,4          TYPE OF LAST CHARACTER
05150  0020 00 4 05200          TRA    RDJT2,4       DISPATCH ON TYPE
                         *
05151  0074 00 4 05233  RDDD    TSX    GET,4         IS A LITERAL STRING
05152  0500 00 0 06403          CLA    GTVAL         USE THIS ITEM AS A DELIMITER
05153  0601 00 0 06400          STO    RDDDC
05154  0074 00 4 05233  RDDDL   TSX    GET,4         GET NEXT CHARACTER
05155  0500 00 0 06400          CLA    RDDDC         GET DELIMITER
05156  0340 00 0 06403          CAS    GTVAL         COMPARE WITH CHARACTER JUST READ
05157  0020 00 0 05161          TRA    *+2           NO
05160  0020 00 0 05203          TRA    RDXT          YES, EXIT
```

```
05161  0074 00 4 05403          TSX    PUT,4       NO, PUT AWAY THE CHARACTER
05162  0020 00 0 05154          TRA    RDDDL       GET NEXT CHARACTER
                        *
05163  0055 00  000002  RDLT    SIR    2           SET LITERAL INDICATOR
05164  0055 00  000001  RDNM    SIR    1           SET FIRST CHARACTER INDICATOR
05165  0074 00 4 05403  RDNN    TSX    PUT,4       PUT THE CHARACTER AWAY
05166  0074 00 4 05233          TSX    GET,4       GET NEXT CHARACTER
05167  0734 00 4 00000          PAX    0,4
05170  0020 00 4 05200          TRA    RDJT2,4     DISPATCH ON TYPE
05171  0020 00 0 05165          TRA    RDNN        $
05172  0020 00 0 05165          TRA    RDNN        LITERAL
05173  0020 00 0 05165          TRA    RDNN        NUMBER
05174  0020 00 0 05203          TRA    RDXT        ,
05175  0020 00 0 05201          TRA    RDPS        (
05176  0020 00 0 05201          TRA    RDPS        )
05177  0020 00 0 05225          TRA    RDPD        .
05200  0020 00 0 05101  RDJT2   TRA    RDJT1       ILLEGAL CHARACTER
                        *
05201  0500 00 4 05046  RDPS    CLA    RDVAL,4     SETUP RDLST CELL
05202  0601 00 0 06402          STO    RDLST
05203  0534 00 4 05410  RDXT    LXA    PUTMC,4     CHARACTER COUNT
05204 -0754 00 0 00000          PXD    0,0         CLEAR AC
05205  3 00005 4 05223          TXH    TPF,4,5     GO IF LAST WORD COMPLETED
05206  0560 00 0 00140          LDQ    SEVENS      GET 77 S
05207  0522 00 4 05473          XEC    PTSFT-1,4   PROPER SHIFT
05210  0774 00 4 00006          AXT    6,4         RESET CHARACTER COUNT
05211  0634 00 4 05410          SXA    PUTMC,4
05212  0534 00 4 05416          LXA    PUTPC,4     WORD COUNT
05213 -0602 00 4 05501          ORS    RDPNB,4     PUT IN PNAME BUFFER
05214 -0754 00 0 00000          PXD    0,0         CLEAR AC
05215  0622 00 4 05507  TPFA    STD    PUTVL+6,4   CHOP OFF PNMAE SAUSAGE
05216  0500 00 0 05501          CLA    PUTVL       GET VALUE
05217 -0774 00 4 05110          AXC    RDPU,4      SET UP TRANSFER TO EXIT
05220  0056 00  000002          RNT    2           TEST LITERAL INDICATOR
05221  0020 00 0 06531          TRA    $NUTRN      MAKE IT A NUMBER
05222  0020 00 0 06406          TRA    INTRN1      MAKE IT AN OBJECT
                        *
05223  0534 00 4 05416  TPF     LXA    PUTPC,4     CORRECT PART COUNT
05224  1 00001 4 05215          TXI    TPFA,4,1
                        *
05225  0054 00  000002  RDPD    RFT    2           TEST FOR LITERAL
05226  0020 00 0 05201          TRA    RDPS        FIRST . TERMONATES A LITERAL
05227  0054 00  000020          RFT    20          TEST FOR FIRST DOT IN A NUMBER
05230  0020 00 0 05201          TRA    RDPS        SECOND . TERMINATES A NUMBER
05231  0055 00  000020          SIR    20          SET DOT INDICATOR
05232  0020 00 0 05165          TRA    RDNN
                        *
05233  0634 00 4 05264  GET     SXA    GTX,4       SAVE LINK IR
05234  0520 00 0 05046          ZET    LRCIS       TEST FOR NEW CARD NEEDED
05235  0020 00 0 05314          TRA    GTGCD       GET A NEW CAERD
05236 -0754 00 0 00000  GETGO   PXD    0,0         CLEAR AC
05237  0560 00 1 06377          LDQ    CELL,1      GET NEXT WORD
05240 -0763 00 0 00003          LGL    3           HIGH ORDER BITS
05241  0734 00 4 00000          PAX    0,4
05242 -0763 00 0 00003          LGL    3           CHARACTER
```

```
05243  0340 00 0 15267          CAS     =057
05244  0020 00 0 05246          TRA     *+2
05245  0020 00 0 05277          TRA     GTPC1       SETUP TO SKIP REST OF WORD,LINE-MARKED
05246  0340 00 0 15271          CAS     =077
05247  0020 00 0 05251          TRA     *+2
05250  0020 00 0 05277          TRA     GTPC1       SKIP WHOLE WORD IN LINE-MARKED FILE
                       *                           ALSO IN MODIFIED CARD-IMAGE FILES
05251  0601 00 0 06403          STO     GTVAL           VALUE OF GET FOR PUT
05252 -0320 00 0 15252          ANA     =7              MASK OUT HIGH ORDER BIT
05253  0621 00 0 05257          STA     GTPT
05254 -0600 00 1 06377          STQ     CELL,1          UPDATE WORD
05255 -2 00001 2 05266          TNX     GTPC,2,1        UPDATE PART COUNT
05256  0560 00 4 05516  GTMC    LDQ     GTTBL,4         GET TABLE ENTRY
05257 -0763 00 0 00000  GTPT    LGL     **              SHIFT PROPER ITEM TO AC
05260  0522 00 0 05257          XEC     GTPT
05261  0522 00 0 05257          XEC     GTPT
05262 -0754 00 0 00000          PXD     0,0             CLEAR AC
05263 -0763 00 0 00003          LGL     3               TYPE NOW IN AC
05264  0774 00 4 00000  GTX     AXT     **,4            RESTORE LINK IR
05265  0020 00 4 00001          TRA     1,4
                       *
05266  0774 00 2 00006  GTPC    AXT     6,2             RELOAD PART COUNT
05267  2 00001 1 05256          TIX     GTMC,1,1        GO IF NEW WORD NOT NEEDED
05270 -0625 00 0 05046          STL     LRCIS           GET NEW CARD
05271  0500 00 0 15274          CLA     =432
05272  0520 00 0 01215          ZET     RDINDC          WHAT IS CURRENT DEVICE
05273  0500 00 0 15255          CLA     =12             DEVICE IS TY
05274  0734 00 1 00000          PAX     0,1             INITIALIZE IR1
05275  0020 00 0 05256          TRA     GTMC                GO BACJ
05276  0 00000 0 00000  PRDIND  PZE                     PREVIOUS DEVICE INDICATOR,ZERO IS DISK
05277  0534 00 2 05370  GTPC1   LXA     CILMAR,2
05300  3 00001 2 05307          TXH     GTPC2,2,1       IS CARD-IMAGE FILE
05301  0500 00 0 15247          CLA     =4              INDICATE CHARACTER IS BLANK
05302  0774 00 2 00006          AXT     6,2
05303  2 00001 1 05264          TIX     GTX,1,1         GET OUT OF GET
05304 -0625 00 0 05046          STL     LRCIS           INDICATE NEW CARD NEEDED
05305  0774 00 1 00660          AXT     432,1
05306  0020 00 0 05264          TRA     GTX
05307  0774 00 2 00006  GTPC2   AXT     6,2
05310  2 00001 1 05234          TIX     GET+1,1,1       GET CHARACTER FROM NEXT WORD
05311 -0625 00 0 05046          STL     LRCIS           NEED NEW RECORD
05312  0774 00 1 00660          AXT     432,1
05313  0020 00 0 05234          TRA     GET+1
                       *
                       * CHANGES FOR NEW FILE SYSTEM BY THE MAN
05314  0520 00 0 01215  GTGCD   ZET     $RDINDC         READING TYPEWRITER
05315  0020 00 0 01211          TRA     $RDFLX1         YES,GO TO IT
05316                           CALLIO  RDWAIT,FILNAM,(=0,(LWPO,,432),GTEOF,EOFCT,DISKER)
```

```
                                    INSERT  $GTGCDF
                                    *
                                    * THIS ROUTINE IS USED TO PREPROCESS CARD-IMAGE FILES
                                    *LINE-MARKED FILES ARE AUTOMATICALLY HANDLED BY THE GET ROUTINE
                                    *
05327  0600 00 0 05276  GTGCD1 STZ     PRDIND          INDICATE PREVIOUS DEVICE WAS DISK
05330  0520 00 0 05370          ZET     CILMAR          HAVE WE DETERMINED LINE-MARKED OR CARD
05331  0020 00 0 05341          TRA     CILA            YES,WE HAVE
05332 -0500 00 0 05517          CAL     LWPO            PICK UP FIRST WORD OF INPUT
05333 -0320 00 0 05367          ANA     CIL77           FIRST CHARACTER 77 TEST
05334  0322 00 0 05367          ERA     CIL77
05335 -0100 00 0 05344          TNZ     CILCI           NO,MUST BE CARD-IMAGE
05336  0500 00 0 15245  CILD    CLA     =1              INDICATE LINE-MARKED
05337  0601 00 0 05370          STO     CILMAR
05340  0020 00 0 05372          TRA     GTEOF-2         MAY NEED MORE RECORDS LATER
05341  0534 00 4 05370  CILA    LXA     CILMAR,4        WHAT TYPE OF FILE ARE WE READING
05342 -3 00001 4 05336          TXL     CILD,4,1        FILE IS LINE-MARKED
05343  0020 00 0 05347          TRA     CILB            FILE IS CARD-IMAGE
05344  0500 00 0 15246  CILCI   CLA     =2
05345  0601 00 0 05370          STO     CILMAR
05346  0600 00 0 05371          STZ     MKWD            CORRECTION FOR CARD IS ZERO
05347  0500 00 0 05371  CILB    CLA     MKWD
05350  0734 00 4 00000          PAX     0,4
05351  1 00642 4 05352          TXI     *+1,4,418
05352  0500 00 0 05367  CILLP   CLA     CIL77
05353  0601 00 4 06375          STO     CELL-2,4
05354  0500 00 0 05367          CLA     CIL77           SETUP TO SKIP THIS WORD COMPLETELY
05355  0601 00 4 06376          STO     CELL-1,4
05356  1 77762 4 05357          TXI     *+1,4,-14
05357  3 00764 4 05361          TXH     *+2,4,500       VALUE OF INDEX LESS THAN ZERO
05360  0020 00 0 05352          TRA     CILLP
05361  0500 00 0 05371          CLA     MKWD
05362 -0100 00 0 05364          TNZ     *+2
05363  0500 00 0 15256          CLA     =14
05364  0402 00 0 15246          SUB     =2
05365  0601 00 0 05371          STO     MKWD
05366  0020 00 0 05372          TRA     GTEOF-2
05367 -370000000000      CIL77  OCT     770000000000
05370  0 00000 0 00000   CILMAR
05371  0 00000 0 00000   MKWD   PZE                     CONTAINS CORRECTION FOR CARD-IMAGE
                         *END OF CHANGES
05372  0600 00 0 05046          STZ     LRCIS               SET SWITCH THAT CARD IS PRESENT
05373  0020 00 0 05236          TRA     GETGO               NO GO ON
                                 *
05374  0535 00 4 05402  GTEOF   LAC     EOFCT,4         PUT $EOF$ DOWN JUST PAST LAST WORD
05375 -0500 00 0 15306          CAL     =H $EOF$        THAT IS, AT LWPO+EOFCT
05376  0602 00 4 05517          SLW     LWPO,4          ..
05377 -0500 00 0 15305          CAL     =H )))))        THIS SHOULD LEAD TO NO PROBLEMS***
05400  0602 00 4 05520          SLW     LWPO+1,4
05401  0020 00 0 05327          TRA     GTGCD1          NOW GO ON AS IF NOTHING HAD HAPPENED

05402  0 00000 0 00000   EOFCT                          SUPPLIED BY RDWAIT AFTER INCOMPLETE REC
                                 *
05403  0054 00  000040   PUT    RFT     40                  TEST TO SEE IF TOOMUCH PNAME
05404  0020 00 0 05432          TRA     PTTFA               GO TO ERROR COMMENT
```

```
05405  0634 00 4 05420          SXA     PUTX,4          SAVE LINK IR
05406  0056 00  000010          RNT     10              TEST FOR FIRST TIME THRU
05407  0020 00 0 05437          TRA     PUTZB           ZERO PNAME BUFFER
05410  0774 00 4 00006  PUTMC   AXT     6,4             CHARACTER COUNT
05411  0500 00 0 06403          CLA     GTVAL           GET CHARACTER
05412  0560 00 0 15244          LDQ     =0
05413  0522 00 4 05474          XEC     PTSFT,4         PROPER SHIFT TO CHARACTER
05414 -2 00001 4 05422          TNX     PTRFP,4,1       DECREMENT CHARACTER COUNT
05415  0634 00 4 05410          SXA     PUTMC,4         UPDATE COUNT CELL
05416  0774 00 4 00005  PUTPC   AXT     5,4             NUMBER OF WORDS IN PNAME
05417 -0602 00 4 05501          ORS     RDPNB,4         PUT CHARACTER IN
05420  0774 00 4 00000  PUTX    AXT     **,4            RESTORE LINK IR
05421  0020 00 4 00001          TRA     1,4             EXIT
05422  0774 00 4 00006  PTRFP   AXT     6,4             RELOAD PART COUNT
05423  0634 00 4 05410          SXA     PUTMC,4
05424  0534 00 4 05416          LXA     PUTPC,4         WORD COUNT
05425 -0602 00 4 05501          ORS     RDPNB,4
05426  2 00001 4 05430          TIX     *+2,4,1         DECREMENT WORD COUNT
05427  0055 00  000040          SIR     40              INDICATE PNAME BUFFER FULL
05430  0634 00 4 05416          SXA     PUTPC,4         UPDATE COUNTER
05431  0020 00 0 05420          TRA     PUTX            GO ON
                        *
05432                    PTTFA  TSSX    WRFLX,4         TOO MANY CHARACTERS
05433  0 00005 0 05474                  RDPNB-5,,5
05434 -0754 00 0 00000          PXD     0,0             CLEAR AC
05435  0074 00 4 01530          TSX     $ERROR,4        GO TO ERROR
05436  452330270300             BCI     1,NCHG30
                        *
05437  0055 00  000010  PUTZB  SIR     10              SET SWITCH
05440  0774 00 4 00005          AXT     5,4             FIX UP BUFFER
05441  0634 00 4 05416          SXA     PUTPC,4         AND PART COUNT
05442  0600 00 4 05501          STZ     RDPNB,4
05443  2 00001 4 05442          TIX     *-1,4,1
05444  0500 00 0 05501          CLA     PUTVL           RELINK THE WORDS
05445  0774 00 4 00005          AXT     5,4
05446  0402 00 0 00126          SUB     $QD1            SET POINTERS
05447  0622 00 4 05507          STD     PUTVL+6,4
05450  2 00001 4 05446          TIX     *-2,4,1
05451  0020 00 0 05410          TRA     PUTMC
                        *
05452 -0625 00 0 05046  TEREAD  STL     LRCIS           SET SWITCH TO GET A NEW CARD
05453  0500 00 0 15251          CLA     =6              SET CELLS
05454  0621 00 0 05065          STA     RDPTS
05455  0621 00 0 05410          STA     PUTMC
05456  0500 00 0 15274          CLA     =432
05457  0621 00 0 05066          STA     RDWDS
05460  0600 00 0 06402          STZ     RDLST
05461 -0754 00 0 00000          PXD     0,0             CLEAR AC
                        * CHANGE TO TEREAD FOR NEW FILE SYSTEM
05462  0600 00 0 05276          STZ     PRDIND
                        *
                        * CALL TP CLOSE AT THIS POINT HAS BEEN DELETED
                        *
05463  0600 00 0 05370          STZ     CILMAR          CLEARS FILE TYPE INFORMATION*****
05464  0600 00 0 01215          STZ     RDINDC          SET DEVICE INDICATION TO DISK
```

```
05465  0020 00 4 00001          TRA     1,4
                        *  END OF CHANGES
                        *
05466 -0763 00 0 00036          LGL     30
05467 -0763 00 0 00030          LGL     24
05470 -0763 00 0 00022          LGL     18
05471 -0763 00 0 00014          LGL     12
05472 -0763 00 0 00006          LGL     6
05473  0761 00 0 00000          NOP
05474                    PTSFT  BSS     0
05501                    RDPNB  BES     5
05501  0 72276 0 00000   PUTVL          ,,-*-1                VALUE OF RDA
05502  0 72275 0 72304                  -RDPNB+5,,-*-1        FOR INTERN OF NUTRN
05503  0 72274 0 72303                  -RDPNB+4,,-*-1
05504  0 72273 0 72302                  -RDPNB+3,,-*-1
05505  0 72272 0 72301                  -RDPNB+2,,-*-1
05506  0 00000 0 72300                  -RDPNB+1
05507 -264430000000             OCT     664430000000,466666660000,660764000000,566666660000
05513 -260126000000             OCT     660126000000,566666660000,550660000000
05516 -155555550000      GTTBL  OCT     555555550000
05517                    LWPO   BSS     2
06377                    CELL   BES     430             NEW SIZE TO HANDLE WHOLE RECORD****
06377                           BSS     1               NEEDED FOR THE PARENS FOLLOWING EOF
                        * HOPE THAT BUFFER DELETED HERE IS NOT USED BY ANYONE*******
06400  0 00000 0 00000   RDDDC
06401  0 00000 0 00000   RDIND                                 INDICATOR STORAGE
06402  0 00000 0 00000  RDLST
06403  0 00000 0 00000  GTVAL
```

LISP FUNCTIONS

```
                                INSERT  $SUBSTF

                                   INTERN

                          I         HED
                          *
                          * INTERN          CHANGED AND MODIFIED TO INCLUDE EXTERNAL ENTRANCES AND
                          *                 THE BUCKET SORT
                          *
06404  0774 00 4 00000    BUCK      AXT     **,4                  BUCJET   THIS MUST BE HERE (INTERN-2)
                          *                                       FOR REMOB
06405 -0600 00 0 06522    BUKSRT    STQ     BSRT                  ATOM TO BE PLACED (CNSFWL ENTRANCE)
06406  0601 00 0 06521    INTRN1    STO     $VALUE                EXTERNAL ENTRANCE FROM APPLY
06407  0634 00 4 06513              SXA     ITRX,4                ENTRANCE FROM READ
06410  0634 00 2 06514              SXA     ITRY,2                SAVE IR S
06411 -0534 00 4 06521              LXD     $VALUE,4              PICK UP POINTER TO PNAME LIST
06412  0500 00 4 00000              CLA     0,4                   GET FIRST WORD OF PNAME
06413  0734 00 4 00000              PAX     0,4
06414 -0500 00 4 00000              CAL     0,4                   GET FIRST WORD IN LOGICAL AC
06415  0765 00 0 00043              LRS     35                    PUT IN MQ AND BIT 35 OF AC
06416  0221 00 0 06523              DVP     BUCKNO                DIVIDE BY NUMBER OF BUCKETS
06417  0760 00 0 00012              DCT                           CHECK DIVISION
06420  0074 00 4 01624              TSX     $DCT,4                DIVIDE ERROR
06421  0734 00 4 00000              PAX     0,4                   REMAINDER TO IR 4
06422  0500 00 4 64457              CLA     BUCKET,4              PICK UP BUCKET
06423  0634 00 4 06404              SXA     BUCK,4                SAVE THE REMAINDER
06424  0734 00 4 00000              PAX     0,4
06425 -0634 00 4 06526              SXD     05,4                  SET UP WORD
06426  0520 00 0 06522              ZET     BSRT                  TEST FOR CNSFWL ENTRANCE
06427  0020 00 0 06516              TRA     INTAD                 YES, GO
06430 -0634 00 4 06524              SXD 01,4
06431 -0534 00 4 06524     04       LXD 01,4                      NEXT OBJECT
06432 -3 00000 4 06472              TXL OUT,4,0                   END OF OBLIST
06433  0500 00 4 00000              CLA ,4
06434  0622 00 0 06524              STD 01
06435  0734 00 4 00000              PAX ,4                        OBJECT M/C NAME
06436 -0634 00 4 06525              SXD 02,4                      PRESERVE IT
06437  0500 00 4 00000              CLA ,4
06440 -0734 00 4 00000     03       PDX ,4                        ADDRESS PART IS -1
06441 -3 00000 4 06431              TXL 04,4,0                    END OF PROPERTY LIST
06442  0500 00 4 00000              CLA ,4
06443  0734 00 4 00000              PAX ,4
06444 -3 11226 4 06440              TXL 03,4,$PNAME-1             NO
06445  3 11227 4 06440              TXH 03,4,$PNAME               NO
06446 -0734 00 4 00000              PDX ,4                        YES IT IS
06447  0500 00 4 00000              CLA ,4
06450  0734 00 4 00000              PAX ,4                        U
06451 -0534 00 2 06521              LXD $VALUE,2                  V
06452 -3 00000 2 06431     07       TXL      04,2,0
06453  0500 00 4 00000              CLA      0,4
06454  0622 00 0 06530              STD Q4                        CDR(U)
06455  0734 00 4 00000              PAX ,4                        CAR(U)
06456  0500 00 2 00000              CLA ,2
06457  0622 00 0 06527              STD Q2                        CDR(V)
06460  0734 00 2 00000              PAX ,2
```

```
06461  0500 00 4 00000          CLA ,4                  CWR(CAR(U))
06462  0402 00 2 00000          SUB ,2                 -CWR(CAR(V))
06463 -0100 00 0 06431          TNZ 04                 NOT THE SAME,NEXT OBJECT
06464 -0534 00 4 06530          LXD Q4,4               CDR(U)
06465 -0534 00 2 06527          LXD Q2,2
06466  3 00000 4 06452          TXH 07,4,0             IF NOT YET END OF NAME
06467  3 00000 2 06431          TXH 04,2,0             IF U,V OF DIFFERENT LENGTH,NEXT
06470  0500 00 0 06525          CLA 02
06471  0020 00 0 06513          TRA     ITRX
06472  0500 00 0 06521  OUT     CLA $VALUE
06473  0074 00 4 07317          TSX $CP1,4
06474  0560 00 0 15244          LDQ     =0
06475  0074 00 4 02714          TSX $CONS,4
06476  0131 00 0 00000          XCA
06477  0500 00 0 00151          CLA OPNA
06500  0074 00 4 02714          TSX $CONS,4
06501  0131 00 0 00000          XCA                    INTO MQ
06502  0500 00 0 00133          CLA     $DMASK         ATOM SYMBOL
06503  0074 00 4 02714          TSX     $CONS,4        MAKE IT AN ATOM
06504  0560 00 0 06526  INTCN   LDQ     05             LIST OF ATOMS IN BUCKET
06505  0622 00 0 06526          STD     05             SAVE ATOM AS ANSWER
06506  0074 00 4 02714          TSX     $CONS,4        ATTACH TO BEGINNING OF LIST
06507  0771 00 0 00022          ARS     18             PUT IN ADDRESS
06510  0522 00 0 06404          XEC     BUCK           BUCKET NUMBER
06511  0621 00 4 64457          STA     BUCKET,4       PUT IN PROPER BUCJET
06512  0500 00 0 06526          CLA     05             ATOM AS ANSWER
06513  0774 00 4 00000  ITRX    AXT     **,4           RESTORE LINK IR
06514  0774 00 2 00000  ITRY    AXT     **,2
06515  0020 00 4 00001          TRA     1,4            EXIT
06516  0500 00 0 06522  INTAD   CLA     BSRT           PICK UP ATOM
06517  0600 00 0 06522          STZ     BSRT           ZERC LOCATICN
06520  0020 00 0 06504          TRA     INTCN          PLACE ATOM IN BICKET
06521  0 00000 0 00000  VALUE                          POINTER TO PNAME LIST
06522  0 00000 0 00000  BSRT                           ATOM ON CNSFWL WENTRANCE
06523  0 00000 0 00177  BUCKNO PZE     127            NUMBER OF BUCKETS
                        *
06524  0 00000 0 00000  01
06525  0 00000 0 00000  02
06526  0 00000 0 00000  05
                 00151  OPNA   SYN     PNAMED
06527  0 00000 0 00000  Q2
06530  0 00000 0 00000  Q4

                        T       HED
06531  0634 00 4 06604  NUTRN   SXA     NX4,4          SAVE INDEX REGISYERS
06532  0634 00 2 06603          SXA     NX2,2
06533  0634 00 1 06602          SXA     NX1,1
06534  0774 00 1 00006          AXT 6,1
06535 -0534 00 4 06521          LXD $VALUE,4
06536  0500 00 4 00000  NA1     CLA 0,4
06537 -0734 00 4 00000          PDX 0,4
06540  0734 00 2 00000          PAX 0,2
06541  0500 00 2 00000          CLA 0,2
06542  0601 00 1 07307          STO BUFFRE+6,1
06543 -3 00000 4 06553          TXL NA2,4,0
```

```
06544   2 00001 1 06536          TIX NA1,1,1
06545  -0634 00 4 01530    NE    SXD $ERROR,4
06546                        TSSX    WRFLX,4
06547   0 00016 0 06360              I$CELL-15,,14
06550  -0754 00 0 00000          PXD    0,0          CLEAR AC
06551   0074 00 4 01531          TSX $ERROR+1,4
06552   454463223127             BCI    1,NMTBIG     NUMBER TOO LARGE IN CONVERSION
06553   0500 00 0 15311    NA2   CLA =H
06554   0601 00 1 07310          STO BUFFRE+7,1
06555   0500 00 0 06606          CLA    KBPOS        PARAMETER FOR NUMBR
06556   0074 00 4 06607          TSX    $NUMBR,4     NUMBER TO MQ
06557   0100 00 0 06545          TZE    NE           OUT-OF-RANGE ERROR
06560  -0120 00 0 06577          TMI    NA7          TRA IF FLOATING NUMBER
06561  -0760 00 0 00001          PBT                 TEST FOR OCTAL NUMBER
06562   0020 00 0 06566          TRA    NA3          TRA IF OCTAL
06563   0500 00 0 00150          CLA    $OCTD        OCTAL SIGNAL FOR $MKNO
06564   0131 00 0 00000          XCA
06565   0020 00 0 06601          TRA    NA8
06566   0131 00 0 00000    NA3   XCA                 NUMBER TO AC
06567   0560 00 0 00142          LDQ    $FIXD        FIX TO MQ
06570  -0120 00 0 06601          TMI    NA8
06571   0340 00 0 00074          CAS    $Q10         TEST FOR 0 THRU 9
06572   0020 00 0 06601          TRA    NA8
06573   0020 00 0 06601          TRA    NA8
06574   0361 00 0 00166          ACL    $HOOA        FORM PRINT OBJECT
06575   0767 00 0 00022          ALS    18
06576   0020 00 0 06602          TRA    NX1
06577   0500 00 0 00143    NA7   CLA    FLOATD       FLOAT SIGNAL FOR $MKNO
06600   0131 00 0 00000          XCA                 NUMBER TO AC
06601   0074 00 4 11163    NA8   TSX    $MKNO,4      MAKE A NUMBER
06602   0774 00 1 00000    NX1   AXT    **,1         RESTORE INDEX REGISTERS
06603   0774 00 2 00000    NX2   AXT    **,2
06604   0774 00 4 00000    NX4   AXT    **,4
06605   0020 00 4 00001          TRA    1,4
06606   0 00001 0 07301    KBPOS PZE    BUFFRE,,1
```

```
                         F      HED
```

NUMBR CONVERTS PACKED BCD CHARACTERS TO A NUMBER WHICH
APPEARS IN MQ. DBC CONVENTIONS ARE FOLLOWED. OCTAL
NUMBERS ARE SIGNALLED BY Q AND MAY BE FOLLOWED BY A
SCALE FACTOR.

ROUTINE STOLEN FROM UADBCI

```
06607   0634 00 1 07134    NUMBR SXA PX1,1            SAVE INDEX REGISTERS
06610   0634 00 2 07135          SXA PX2,2
06611   0634 00 4 07136          SXA PX4,4
06612   0602 00 0 00047          SLW T
06613   0737 00 2 00000          PAC ,2               IR2 HAS WORD COUNT
06614  -0737 00 1 00000          PDC ,1               IR1 WILL GET CHARACTER COUNT
06615   0771 00 0 00021          ARS 17
06616   0601 00 0 00050          STO N
```

```
06617  0767 00 0 00001          ALS 1
06620  0400 00 0 00050          ADD N
06621  0737 00 4 00000          PAC ,4
06622  0560 00 2 00000          LDQ 0,2          PUT BCD WORD IN MQ
06623 -0763 00 4 77772          LGL -6,4         SHIFT OUT EXTRA CHARACTERS
06624 -0600 00 0 00044          STQ MQ           SAVE FIRST BATCH OF CHARACTERS
06625  1 00007 1 06626          TXI *+1,1,7

                                LOOK AT CHARACTERS UNTIL A Q OR NON-OCTAL CHARACTER APPEARS

06626 -0754 00 0 00000  CY3     PXD     ,0
06627 -0763 00 0 00006          LGL 6
06630  0402 00 0 15253          SUB =8           TEST FOR OCTAL DIGIT
06631  0120 00 0 06636          TPL CY4
06632  2 00001 1 06626  CY2     TIX CY3,1,1      GET NEXT CHARACTER
06633  1 77777 2 06634          TXI *+1,2,-1
06634  0560 00 2 00000          LDQ 0,2
06635  1 00005 1 06626          TXI CY3,1,5
06636  0400 00 0 15253  CY4     ADD =8
06637  0340 00 0 15265          CAS =H00000Q
06640  0020 00 0 06650          TRA DECNO
06641  0020 00 0 07147          TRA OCTNO        IF Q, NUMBER IS OCTAL
06642  0340 00 0 15264          CAS =H00000-       IF CHARACTER IS MINUS, PLUS, OR DASH,
06643  0020 00 0 06650          TRA DECNO        LOOK AT MORE CHARACTERS,
06644  0020 00 0 06632          TRA CY2          OTHERWISE NUMBER IS DECIMAL
06645  0340 00 0 15257          CAS =H00000+
06646  0020 00 0 06650          TRA DECNO
06647  0020 00 0 06632          TRA CY2

06650  0535 00 2 00047  DECNO   LAC T,2           IR2 HAS WORD COUNT
06651 -0535 00 1 00047          LDC T,1          IR1 WILL GET CHARACTER COUNT
06652  0560 00 0 00044          LDQ MQ           RESTORE FIRST GROUP OF CHARACTERS
06653 -0754 00 0 00000          PXD     ,0
06654  0602 00 0 00044  BN2     SLW BN           REGISTERS
06655  0602 00 0 00045  EX2     SLW EXPN
06656  0602 00 0 00050  INTN    SLW N
06657  0774 00 4 00000          AXT     0,4        XR4 IS DECIMAL COUNT
06660 -0500 00 0 06750          CAL SW1          RESET SWITCHES FOR
06661  0630 00 0 07012          STP CM2           FIXED POINT
06662  0630 00 0 07065          STP CM6           X
06663  0630 00 0 06770          STP EXS           EXP
06664  0630 00 0 07014          STP CM3           POINT
06665  0630 00 0 06777          STP CX3           DECIMAL NUMBER
06666 -0500 00 0 06656          CAL INTN         INITIALIZE CONVERSION
06667  1 00010 1 06702          TXI BN3,1,8      FIX INITIAL CHARACTER COUNT
06670  0502 00 0 07014  PT1     CLS CM3          INVERT SWITCH TO SIGNAL DECIMAL POINT
06671  0601 00 0 07014          STO CM3
06672 -0500 00 0 06725          CAL CV3
06673  0621 00 0 06742          STA CV5          ROUTINE TO COUNT
06674  0621 00 0 06745          STA CV6          DECIMAL PLACES
06675  1 00001 4 06742          TXI CV5,4,1
06676  1 77777 4 06725  PT3     TXI CV3,4,-1     COUNT DECIMAL PLACES
06677  0502 00 0 06770  EX1     CLS EXS          INVERT SWITCH TO SIGNAL EXPONENT
06700  0601 00 0 06770          STO EXS
06701 -0500 00 0 06655          CAL EX2          SET UP EXPONENT CONVERSION
```

LISP FUNCTIONS

```
      06702  0621 00 0 06733   BN3   STA CV7          STORE CONVERSION
      06703  0621 00 0 06735         STA CV8          ADDRESS
      06704  0621 00 0 06741         STA CV9
      06705 -0500 00 0 06676         CAL PT3          INITIAL CONVERSION
      06706  0621 00 0 06742         STA CV5          WITHOUT DECIMAL COUNT
      06707  0621 00 0 06745         STA CV6
      06710 -0500 00 0 06735   PL1   CAL CV8
      06711  0622 00 0 06737   MN3   STD CV10
      06712  0140 00 0 06742         TOV CV5
TD    06713 -3 00000 0 06742         TXL CV5
      06714  0500 00 0 07012   BN1   CLA CM2          INVERT SWITCHES TO SIGNAL FIXED POINT
      06715  0630 00 0 07012         STP CM2
      06716  0630 00 0 07065         STP CM6
      06717 -0500 00 0 06654         CAL BN2          SET UP B CONVERSION
TD    06720 -3 00000 0 06702         TXL BN3
      06721  0500 00 0 00131   MN1   CLA PBIT         START NEGATIVE ACCUMULATION WITH NEG. ZERO
      06722  0601 60 0 06733         STO* CV7
      06723 -0500 00 0 06724         CAL MN2          OP CODE TO MAKE CV10 A SUB INSTRUCTION
      06724 -3 40200 0 06711   MN2   TXL MN3,0,258*64
      06725 -0754 00 0 06676   CV3   PXD     PT3,0
      06726 -0763 00 0 00006         LGL 6
      06727  0340 00 0 15254         CAS =10          TEST FOR DIGIT
TD    06730 -3 00000 0 06747         TXL CM
TD    06731 -3 00000 0 06766         TXL CV2
      06732  0602 00 0 00046         SLW CH           PERFORM CODED
      06733  0500 00 0 00050   CV7   CLA N            MULTIPLICATION
      06734  0767 00 0 00002         ALS 2            BY TEN AND ADD
      06735  0400 00 0 00050   CV8   ADD N            CURRENT DIGIT
      06736  0767 00 0 00001         ALS 1
      06737  0400 00 0 00046   CV10  ADD CH
      06740  0140 00 0 06746         TOV OVF          TEST FOR OVERFLOW
      06741  0601 00 0 00050   CV9   STO N
      06742  2 00001 1 06725   CV5   TIX CV3,1,1      COUNT CHARACTERS
      06743  1 77777 2 06744         TXI CV4,2,-1     OBTAIN NEXT BCD
      06744  0560 00 2 00000   CV4   LDQ 0,2          WORD AND RESTORE
      06745  1 00005 1 06725   CV6   TXI CV3,1,5      CHARACTER COUNT
      06746  1 00001 4 06742   OVF   TXI CV5,4,1      COUNT DECIMAL OVERFLOWS
      06747  0340 00 0 15264   CM    CAS =H00000-
TD    06750 -3 00000 0 06766   SW1   TXL CV2
TD    06751 -3 00000 0 06721         TXL MN1
      06752  0340 00 0 15263         CAS =H00000.
TD    06753 -3 00000 0 06766         TXL CV2
TD    06754 -3 00000 0 06670         TXL PT1
      06755  0340 00 0 15262         CAS =H00000E
TD    06756 -3 00000 0 06766         TXL CV2
TD    06757 -3 00000 0 06677         TXL EX1
      06760  0340 00 0 15260         CAS =H00000B
TD    06761 -3 00000 0 06766         TXL CV2
TD    06762 -3 00000 0 06714         TXL BN1
      06763  0340 00 0 15257         CAS =H00000+
TD    06764 -3 00000 0 06766         TXL CV2
TD    06765 -3 00000 0 06710         TXL PL1
      06766  0500 00 0 00050   CV2   CLA N
      06767  0100 00 0 07140         TZE STZ          SEE IF ZERO FIXED OR FLOATING
TD    06770 -3 00000 0 06777   EXS   TXL CX3          SWITCH - TXH INDICATES EXPONENT
```

```
        06771 -0500 00 0 00131          CAL P8IT            PREPARE TRUE
        06772  0400 00 0 00045          ADD EXPN            DECIMAL EXPONENT
        06773  0767 00 0 00022          ALS 18
        06774  0622 00 0 06776          STD CM4
        06775  0500 00 0 00050          CLA N
        06776  1 00000 4 07015  CM4     TXI     CM5,4,0
TD      06777 -3 00000 0 07012  CX3     TXL CM2             SWITCH - TXH INDICATES OCTAL
                                        SCALE OCTAL NUMBER
        07000  0500 00 0 00044          CLA BN              MULTIPLY SCALE FACTOR BY 3
        07001  0767 00 0 00001          ALS 1                FOR NUMBER OF SHIFTS NEEDED
        07002  0400 00 0 00044          ADD BN
        07003  0621 00 0 07005          STA CX5
        07004  0500 00 0 00050          CLA N
        07005  0767 00 0 00000  CX5     ALS **
        07006 -0760 00 0 00001          PBT                 ALLOW FOR P BIT
        07007  0020 00 0 07105          TRA ISTOR
        07010 -0760 00 0 00003          SSM
        07011  0020 00 0 07105          TRA ISTOR

TD      07012 -3 00000 0 07014  CM2     TXL CM3             SWITCH - INVERTED TO TXH INDICATES FIXED POINT
TD      07013 -3 00000 0 07015          TXL CM5
TD      07014 -3 00000 0 07105  CM3     TXL ISTOR           SWITCH - TXH INDICATES POINT
        07015  0621 00 0 07217  CM5     STA FL1             35 BIT INTEGER
        07016  0771 00 0 00017          ARS 15
        07017 -0501 00 0 07220          ORA FL2
        07020  0300 00 0 07220          FAD FL2
        07021  0120 00 0 07024          TPL CMF1
        07022  0302 00 0 07217          FSB FL1
TD      07023 -3 00000 0 07025          TXL CMF2
        07024  0300 00 0 07217  CMF1    FAD FL1
        07025 -0600 00 0 00052  CMF2    STQ RESID
        07026 -3 00000 4 07065          TXL     CM6,4,0
        07027  3 00046 4 07050          TXH CM7,4,38        TEST FOR NEGATIVE EXP
        07030  0634 00 4 07031          SXA *+1,4           COMPUTE ABSOLUTE VALUE OF EXPONENT
        07031 -0774 00 4 00000          AXC **,4
        07032  0601 00 0 00051          STO DATUM
        07033  0560 00 4 07220          LDQ ONE,4           COMPUTE FLOATING
        07034  0260 00 0 00051          FMP DATUM           BINARY REPRESENTATION
        07035  0601 00 0 00047          STO T               OF INTEGER TIMES THE
        07036 -0600 00 0 00050          STQ T+1             POWER OF TEN GIVEN
        07037  0560 00 4 07220          LDQ ONE,4           BY THE TRUE EXPONENT
        07040  0260 00 0 00052          FMP RESID
        07041  0300 00 0 00050          FAD T+1
        07042  0300 00 0 00047          FAD T
        07043  0361 00 0 07215          ACL EXC1
        07044 -0760 00 0 00001          PBT
TD      07045 -3 00000 0 07065          TXL CM6
        07046 -0754 00 0 00000  CM8     PXD     ,0
        07047  0020 00 0 07134          TRA PX1             NUMBER OUT OF RANGE, EXIT WITH 0 IN AC
        07050 -3 77717 4 07046  CM7     TXL CM8,4,-49       TEST FOR ILLEGAL EXP
        07051  0161 00 0 07052  CM13    TQO CM13+1
        07052  0241 00 4 07220          FDP ONE,4           COMPUTE FLOATING
        07053 -0600 00 0 00047          STQ T               BINARY EQUIVALENT
        07054  0300 00 0 00052          FAD RESID           OF INTEGER TIMES
        07055  0241 00 4 07220          FDP ONE,4           POWER OF TEN GIVEN
```

```
        07056  0161 00 0 07046          TQU CM8
        07057 -0600 00 0 00050          STQ T+1          BY TRUE EXPONENT
        07060  0500 00 0 00050          CLA T+1
        07061  0300 00 0 00047          FAD T
        07062  0361 00 0 07216          ACL EXC2
        07063 -0760 00 0 00001          PBT
TD      07064 -3 00000 0 07046          TXL CM8
TD      07065 -3 00000 0 07110  CM6     TXL FSTOR        SWITCH - TXH INDICATES FIXED POINT
        07066  0601 00 0 00047          STO T
        07067  0767 00 0 00002          ALS 2
        07070 -0760 00 0 00003          SSM              DETERMINE SHIFT
        07071  0771 00 0 00035          ARS 29           NECESSARY TO POSITION
        07072  0400 00 0 15273          ADD      =128
        07073  0400 00 0 00044          ADD BN           BY B
        07074  0120 00 0 07076          TPL SHIFT
        07075 -0100 00 0 07046          TNZ CM8
        07076  0621 00 0 07104  SHIFT   STA CM12
        07077  0500 00 0 00047          CLA T            REMOVE CHARACTERISTICS
        07100  0763 00 0 00010          LLS 8            FROM FLOATING NUMBER
        07101  0767 00 0 00002          ALS 2
        07102  0771 00 0 00012          ARS 10
        07103  0763 00 0 00010          LLS 8
        07104  0765 00 0 00000  CM12    LRS      **
        07105  0131 00 0 00000   ISTOR  XCA              RESULT TO MQ
        07106 -0500 00 0 06750   ISTO1  CAL SW1          SET FIXED POINT INDICATOR SWITCH
        07107  0020 00 0 07112          TRA XT3
        07110  0131 00 0 00000  FSTOR   XCA              RESULT TO MQ
        07111  0500 00 0 06750          CLA SW1          SET FLOAT INDICATOR SWITCH
        07112  0630 00 0 07132  XT3     STP XT1
        07113  2 00001 1 07116          TIX XT2,1,1      IF NO SIGNIFICANT CHARACTERS
        07114  1 77777 2 07115          TXI *+1,2,-1      LEFT IN WORD, MOVE TO  NEXT WORD
        07115  0774 00 1 00006          AXT 6,1
        07116 -0754 00 1 00000  XT2     PXD ,1           SET POSITION INDICATORS
        07117  0402 00 0 00127          SUB QD7
        07120  0602 00 0 00047          SLW T
        07121 -0500 00 0 06777          CAL CX3                  P BIT IN OUTPUT INDICATES OCTAL
        07122 -0320 00 0 00131          ANA $SBIT
        07123 -0602 00 0 00047          ORS T
        07124  0760 00 0 00006          COM
        07125  0630 00 0 00047          STP T
        07126  0634 00 2 07127          SXA *+1,2
        07127 -0774 00 2 00000          AXC **,2
        07130  0754 00 2 00000          PXA ,2
        07131  0361 00 0 00047          ACL T
TD      07132 -3 00000 0 07134  XT1     TXL *+2          SET SIGN + FOR FIXED,
        07133 -0760 00 0 00003          SSM               - FOR FLOATING
        07134  0774 00 1 00000  PX1     AXT ,1           RESTORE INDEX REGISTERS
        07135  0774 00 2 00000  PX2     AXT ,2
        07136  0774 00 4 00000  PX4     AXT ,4
        07137  0020 00 4 00001          TRA 1,4          EXIT

                        WE GET HERE IF NUMBER IS ZERO.
                        WE HERE DECIDE WHETHER WE ARE FACED WITH A FIXED OR FLOATING
                         ZERO.
```

```
07140  0560 00 0 07012   STZ   LDQ CM2              TXH (+) IF B
07141  0162 00 0 07105         TQP ISTOR
07142  0560 00 0 07014         LDQ CM3              TXH (+) IF DECIMAL POINT FOUND
07143  0162 00 0 07110         TQP FSTOR
07144  0560 00 0 06770         LDQ EXS              TXH (+) IF E FOUND
07145  0162 00 0 07110         TQP FSTOR
07146  0020 00 0 07105         TRA ISTOR

                               PROCESS OCTAL NUMBER

07147  0535 00 2 00047   OCTNO LAC T,2              IR2 HAS WORD COUNT
07150 -0535 00 1 00047         LDC T,1              IR1 WILL GET CHARACTER COUNT
07151  0560 00 0 00044         LDQ MQ               RESTORE FIRST GROUP OF CHARACTERS
07152 -0754 00 0 00000         PXD    ,0
07153  0621 00 0 06733         STA CV7              SET SIGNAL FOR OCTAL NUMBER
07154  1 00010 1 07162         TXI OCT9,1,8         FIX CHARACTER COUNT
07155 -0754 00 0 00000   OCT1  PXD    ,0
07156 -0763 00 0 00003         LGL 3
07157 -0100 00 0 07170         TNZ OCT8
07160  0500 00 0 00050         CLA N
07161 -0763 00 0 00003         LGL 3
07162  0601 00 0 00050   OCT9  STO N                ALLOW FOR BOTH P BIT AND MINUS SIGN
07163 -0602 00 0 00050         ORS N
07164  2 00001 1 07155   OCT6  TIX OCT1,1,1
07165  1 77777 2 07166         TXI OCT2,2,-1
07166  0560 00 2 00000   OCT2  LDQ 0,2              NEW PACKED WORD
07167  1 00005 1 07155         TXI OCT1,1,5
07170 -0763 00 0 00003   OCT8  LGL 3
07171  0340 00 0 15265         CAS =H000000                 TEST FOR OCTAL SCALE FACTOR
07172  0020 00 0 07202         TRA OCT3
07173  0020 00 0 07206         TRA OCT10
07174  0340 00 0 15264         CAS     =HC0000-
TD     07175 -3 00000 0 07202  TXL OCT3
TD     07176 -3 00000 0 07204  TXL OCT5
07177  0340 00 0 15257         CAS     =HC0000+
TD     07200 -3 00000 0 07202  TXL OCT3
TD     07201 -3 00000 0 07164  TXL OCT6
07202  0560 00 0 00050   OCT3  LDQ N
TD     07203  1 00000 0 07106  TXI ISTO1
07204  0500 00 0 00131   OCT5  CLA PBIT             SET NEGATIVE SIGN
TD     07205 -3 00000 0 07162  TXL OCT9
07206  0500 00 0 06777   OCT10 CLA CX3              SET SWITCH FOR OCTAL SCALE FACTOR
07207  0630 00 0 06777         STP CX3
07210  0600 00 0 00044         STZ BN               CLEAR SCALE FACTOR CELL
07211 -0500 00 0 06750         CAL SW1              SET EXPONENT SWITCH TO OFF
07212  0630 00 0 06770         STP EXS
07213 -0500 00 0 06654         CAL BN2              SET UP Q CONVERSION
07214  0020 00 0 06702         TRA BN3

                         00131   PBIT  SYN $SBIT
                         00127   QD7   SYN     $QD7
07215 +043000000000       EXC1  DEC 35B8            CHARACTERISTIC=35
07216 +335000000000       EXC2  DEC 221B8           CHAR.=COMPL. 35
```

```
07217 +233000000000        FL1    DEC 15588
07220 +252000000000        FL2    DEC 17088
07221 +141500000000               OCT 141500000000,144620000000,147764000000,153470400000
07225 +156606500000               OCT 156606500000,161750220000,165461132000,170575360400
07231 +173734654500               OCT 173734654500,177452013710,202564416672,205721522451
07235 +211443023471               OCT 211443023471,214553630410,217706576512,223434157116
07241 +226543212741               OCT 226543212741,231674055532,235425434430,240532743536
07245 +243661534466               OCT 243661534466,247417031702,252522640262,255647410336
07251 +261410545213               OCT 261410545213,264512676456,267635456171,273402374714
07255 +276503074077               OCT 276503074077,301623713116,304770675742,310473426555
07261 +313612334311               OCT 313612334311,316755023373,322464114135,325601137164
07265 +330741367021               OCT 330741367021,334454732313,337570120775,342726145174
07271 +346445677216               OCT 346445677216,351557257061,354713132676,360436770626
07275 +363546566774               OCT 363546566774,366700324573,372430204755,375536246150
                     07220 ONE     SYN FL2
07301                      REORG  BSS 0
                     00044         ORG COMMON
00044                      BN     BSS 1
                     00044 MQ      SYN BN
00045                      EXPN   BSS 1
00046                      CH     BSS 1
00047                      CHD    BSS 1
                     00047 T       SYN CHD
00050                      N      BSS 1
00051                      DATUM  BSS 1
00052                      RESID  BSS 1
                     07301         ORG REORG                RESTORE ORIGIN
07301                      BUFFRE BSS 14


                           R      HED

                                  FUNCTION CP1
                                  CP1(L)=(L=0 YIELDS 0,
                                        OTHERWISE CONS(CONSW(CWR(CAR(L))),CP1(CDR(L))))

                           C      HED
07317  0634 00 4 07344     CP1    SXA    CPX,4
07320 -0774 00 4 07346            AXC    CPE,4
07321 -0634 00 4 07346            SXD    CPE,4
07322 -0734 00 4 00000            PDX    0,4
07323  0600 00 0 02571            STZ    CPF
07324 -3 00000 4 07343            TXL    CPA,4,0
07325  0500 00 4 00000     CPL    CLA    0,4
07326  0622 00 0 07347            STD    CPT
07327  0734 00 4 00000            PAX    0,4
07330  0500 00 4 00000            CLA    0,4
07331  0074 00 4 02674            TSX    $CONSW,4
07332  0560 00 0 15244            LDQ    =0
07333  0074 00 4 02714            TSX    $CONS,4
07334 -0520 00 0 02571            NZT    CPF
07335  0601 00 0 02571            STO    CPF
07336 -0534 00 4 07346            LXD    CPE,4
07337  0622 00 4 00000            STD    0,4
07340  0622 00 0 07346            STD    CPE
```

```
07341 -0534 00 4 07347          LXD     CPT,4
07342  3 00000 4 07325          TXH     CPL,4,0
07343  0500 00 0 02571  CPA     CLA     CPF
07344  0774 00 4 00000  CPX     AXT     **,4
07345  0020 00 4 00001          TRA     1,4
07346  0 00000 0 00000  CPE
07347  0 00000 0 00000  CPT
```

SUBST

```
SUBST(L,V,M) =
    (M = 0 YIELDS 0,
    EQUAL(M,V) YIELDS COPY(L),
                      CAR(M)=-1 YIELDS M
    1 YIELDS CONS(SUBST(L,V,CAR(M)),SUBSTL,V,CDR(M))))
```

```
                        R       HED
07350  0601 00 0 02655   SUBST  STO     SX
07351 -0600 00 0 02656          STQ     SY
07352  0500 00 0 02531          CLA     $ARG3
07353 -0634 00 4 02653   SUB1   SXD     SXT,4
07354  0601 00 0 02657          STO     ST
07355  0560 00 0 02656          LDQ     SY
07356  0074 00 4 03450          TSX     $EQUAL,4
07357 -0100 00 0 07424          TNZ     SUB4
07360 -0534 00 4 02657          LXD     ST,4
07361  0500 00 4 00000          CLA     0,4
07362  0734 00 4 00000          PAX     0,4
07363  0500 00 0 02657          CLA     ST
07364  3 77776 4 07422          TXH     SUB2,4,-2
07365  0074 00 4 01723          TSX     $SAVE,4
07366 -3 02656 0 01766          TXL     $END2,,SZ+2
07367  0622 00 0 02654          STD     SZ
07370 -0734 00 4 00000          PDX     0,4
07371  0500 00 4 00000          CLA     0,4
07372 -0734 00 4 00000          PDX     0,4
07373  0634 00 4 02654          SXA     SZ,4
07374  0734 00 4 00000          PAX     0,4
07375 -0754 00 4 00000          PXD     0,4
07376  0074 00 4 07353          TSX     SUB1,4
07377  0534 00 4 02654          LXA     SZ,4
07400  0771 00 0 00022          ARS     18
07401  0621 00 0 02654          STA     SZ
07402 -0754 00 4 00000          PXD     0,4
07403  0074 00 4 07353          TSX     SUB1,4
07404 -0534 00 4 02654          LXD     SZ,4
07405  0622 00 0 02654          STD     SZ
07406  0500 00 4 00000          CLA     0,4
07407  0402 00 0 02654          SUB     SZ
07410  0100 00 0 07420          TZE     SUB3
07411 -0534 00 4 02735          LXD     $FREE,4
07412  3 00000 4 07414          TXH     *+2,4,0
07413  0074 00 4 03037          TSX     $FROUT,4
07414  0500 00 4 00000          CLA     0,4
07415  0622 00 0 02735          STD     $FREE
```

LISP FUNCTIONS

```
07416  0500 00 0 02654          CLA     SZ
07417  0601 00 4 00000          STO     0,4
07420 -0754 00 4 00000  SUB3    PXD     0,4
07421  0074 00 4 01736          TSX     UNSAVE,4
07422 -0534 00 4 02653  SUB2    LXD     SXT,4
07423  0020 00 4 00001          TRA     1,4
07424  0500 00 0 02655  SUB4    CLA     SX
07425  0020 00 0 07422          TRA     SUB2
```

```
                                    INSERT  $APENDF

                                        APPEND(L1,L2)=
                                        (L1=0 YIELDS L2,1 YIELDS CONS(CAR(L1),APPEND(XDR(L1),L2)))
                                A       HED
     07426 -0100 00 0 07431     APPEND  TNZ     APNP1
     07427  0131 00 0 00000             XCA
     07430  0020 00 4 00001             TRA     1,4
     07431 -0634 00 4 02556     APNP1   SXD     AS1,4
     07432  0074 00 4 01723             TSX     $SAVE,4
     07433 -3 02561 0 01766             TXL     $END2,,CWR1+2       SAVE 2 ITEMS
     07434 -0734 00 4 00000             PDX     0,4
     07435  0500 00 4 00000             CLA     0,4
     07436  0601 00 0 02557             STO     CWR1
     07437 -0320 00 0 00133             ANA     DECM
     07440  0074 00 4 07426             TSX     APPEND,4
     07441  0131 00 0 00000             XCA
     07442  0534 00 4 02557             LXA     CWR1,4
     07443 -0754 00 4 00000             PXD     0,4
     07444  0074 00 4 01736             TSX     UNSAVE,4
     07445 -0534 00 4 02556             LXD     AS1,4
     07446  0020 00 0 02714             TRA     $CONS
                       00133    DECM    SYN     $DMASK

                                        PAIR
                                *           RECODED TO MAKE LISTS IN DOT NOTATION

                                A       HED
     07447  0634 00 4 07455     PAIR    SXA     PAIRX,4         SAVE LINK IR
     07450 -0600 00 0 02632             STQ     LIS             ARG 2
     07451  0560 00 0 07457             LDQ     FARG            PICK UP FUNCTIONAL ARGUMENT
     07452  0074 00 4 03201             TSX     MAPLIS,4        LET MAPLIST DO THE CONSING
     07453  0520 00 0 02632             ZET     LIS             TEST FOR ARG 2 GONE TO END
     07454  0020 00 0 07477             TRA     PERF            DID NOT, GO TO ERROR
     07455  0774 00 4 00000     PAIRX   AXT     **,4            RESTORE LINK IR
     07456  0020 00 4 00001             TRA     1,4             EXIT
                                *
     07457 -3 00001 0 07460     FARG    TXL     *+1,,1          PAIR FUNCTIONAL ARGUMENT FOR MAPLIST
     07460  0634 00 4 07475             SXA     FARGX,4         SAVE LINK IR
     07461  0622 00 0 02631             STD     TEM             SAVE ARGUMENT
     07462 -0534 00 4 02632             LXD     LIS,4           PICK UP 2ND ARG LIST
     07463 -3 00000 4 07502             TXL     PERS,4,0        GO IF NO MORE 2ND ARG
     07464  0500 00 4 00000             CLA     0,4             NEXT WORD
     07465  0734 00 4 00000             PAX     0,4             CAR
     07466  0622 00 0 02632             STD     LIS             SAVE CDR
     07467 -0754 00 4 00000             PXD     0,4             CAR INTO DECREMENT
     07470  0131 00 0 00000             XCA                     INTO MQ
     07471 -0534 00 4 02631             LXD     TEM,4           LIST 1
     07472  0500 00 4 00000             CLA     0,4             TAKE CAR OF LIST
     07473  0734 00 4 00000             PAX     0,4
     07474 -0754 00 4 00000             PXD     0,4
     07475  0774 00 4 00000     FARGX   AXT     **,4            RESTORE LINK IR
     07476  0020 00 0 02714             TRA     $CONS
                                *           FIRST ARG LIST TOO SHORT ERROR
     07477                      PERF    ERROR   A1LSA2
```

```
                          *                ERROR, SECOND ARG LIST TOO SHORT
07502                     PERS    ERROR    A2LSA1
                          *
                          *
                          *
```

```
                                   MAPCAR(L,F) = (L=0 YIELDS 0,
                                                  F(L) YIELDS 0,
                                                  1 YIELDS MAPCAR(CDR(L),F))
```

```
                          D       HED
07505  0100 00 4 00001    MAPCAR  TZE 1,4
07506 -0634 00 4 02612             SXD RET,4
07507  0074 00 4 01723             TSX $SAVE,4
07510 -3 02616 0 01764             TXL     $END3,,F+2          SAVE 3 ITEMS
07511 -0600 00 0 02614             STQ F
07512  0601 00 0 02613    MCPR    STO L
07513 -0534 00 4 02614             LXD F,4
07514  3 00012 4 07517             TXH *+3,4,10
07515  0074 00 4 02614             TSX F,4
07516  0020 00 0 07522             TRA *+4
07517 -0634 00 4 07521             SXD *+2,4
07520  0074 00 4 10565             TSX COMPAT,4
07521  0 00000 0 00001                 1,,**
07522 -0534 00 4 02613             LXD L,4
07523  0500 00 4 00000             CLA 0,4
07524 -0734 00 4 00000             PDX ,4
07525 -0754 00 4 00000             PXD ,4
07526 -0100 00 0 07512             TNZ     MCPR
07527  0074 00 4 01736             TSX UNSAVE,4
07530 -0534 00 4 02612             LXD RET,4
07531  0020 00 4 00001             TRA 1,4
                                       MAPCON(L,F)=
                                       (L=0 YIELDS 0,,1 YIELDS NCONC(F(L),MAPCON(CDR(L),F)))
                          R       HED
07532  0100 00 4 00001    MAPCON  TZE 1,4
07533 -0634 00 4 02615             SXD MCN5,4
07534  0074 00 4 01723             TSX $SAVE,4
07535 -3 02622 0 01762             TXL     $END4,,MCN2+2       SAVE 4 ITEMS
07536  0601 00 0 02617             STO MCN3
07537 -0600 00 0 02616             STQ MCN4
07540 -0534 00 4 02616             LXD MCN4,4
07541  3 00012 4 07544             TXH *+3,4,10
07542  0074 00 4 02616             TSX MCN4,4
07543  0020 00 0 07547             TRA *+4
07544 -0634 00 4 07546             SXD *+2,4
07545  0074 00 4 10565             TSX COMPAT,4
07546  0 00000 0 00001                 1,,**
07547  0601 00 0 02620             STO MCN2
07550 -0534 00 4 02617             LXD MCN3,4
07551  0500 00 4 00000             CLA 0,4
07552 -0320 00 0 00133             ANA MCDM
07553  0560 00 0 02616             LDQ MCN4
07554  0074 00 4 07532             TSX MAPCON,4
07555  0131 00 0 00000             XCA
```

```
07556   0500 00 0 02620          CLA MCN2
07557   0074 00 4 01736          TSX UNSAVE,4
07560  -0534 00 4 02615          LXD MCN5,4
07561   0020 00 0 07562          TRA     $NCONC
                  00133   MCDM   SYN     $DMASK
                                 FUNCTION NCONC
                                 /     L1=0 YIELDS RETURN(L2)
                                       M=L1
                                 A2    CDR(M)=0 YIELDS GO A1
                                       M=CDR(M)
                                       GO A2
                                 A1    CDR(M)=L2
                                 //    RETURN(L1)
                           R     HED
07562  -0100 00 0 07565   NCONC TNZ NCI1
07563   0131 00 0 00000          XCA
07564   0020 00 4 00001          TRA 1,4
07565   0634 00 4 07576   NCI1   SXA     NCS1,4              SAVE LINK IR
07566   0601 00 0 07600          STO NCS3
07567  -0734 00 4 00000    NCI2  PDX 0,4
07570   0500 00 4 00000          CLA 0,4
07571  -0320 00 0 00133          ANA NCDM
07572  -0100 00 0 07567          TNZ NCI2
07573   0131 00 0 00000          XCA
07574   0622 00 4 00000          STD 0,4
07575   0500 00 0 07600          CLA NCS3
07576   0774 00 4 00000   NCS1   AXT     **,4                RESTORE LINK IR
07577   0020 00 4 00001          TRA 1,4
                  00133   NCDM   SYN     $DMASK
07600   0 00000 0 00000    NCS3
                                 REMPRP REMOVES THE PROPERTY GIVEN BY THE MQ FROM THE
                                 OBJECT GIVEN BY THE AC
07601  -0634 00 4 07631  REMPRP SXD RMPRX,4
07602  -0600 00 0 02530          STQ $ARG2
07603  -0534 00 4 02530          LXD $ARG2,4
07604  -0634 00 4 07617          SXD RMPRT+1,4
07605   1 77777 4 07606          TXI **+1,4,-1
07606  -0634 00 4 07616          SXD RMPRT,4
07607  -0734 00 4 00000          PDX 0,4
07610   0020 00 0 07613          TRA RMPR2
07611   0500 00 0 02530   RMPR1  CLA $ARG2
07612   0601 00 0 02531          STO $ARG3
07613  -0634 00 4 02530   RMPR2  SXD $ARG2,4
07614   0500 00 4 00000          CLA 0,4
07615   0734 00 4 00000          PAX 0,4
07616  -3 00000 4 07620   RMPRT  TXL **+2,4,**
07617  -3 00000 4 07624          TXL RMPRE,4,**
07620  -0734 00 4 00000          PDX 0,4
07621   3 00000 4 07611          TXH RMPR1,4,0
07622  -0534 00 4 07631          LXD RMPRX,4
07623   0020 00 4 00001          TRA 1,4
07624  -0734 00 4 00000   RMPRE  PDX 0,4
07625   0500 00 4 00000          CLA 0,4
07626  -0534 00 4 02531          LXD $ARG3,4
07627   0622 00 4 00000          STD 0,4
```

LISP FUNCTIONS

```
07630   0020 00 0 07613          TRA RMPR2
07631   0 00000 0 00000    RMPRX

                        ****** PRINAR HAS BEEN DELETED*******
                        ****************************************
```

PROP AND SASSOC
SPECIALIZED SEARCH ROUTINES WHICH SHARE STORAGE

R HED

```
                                PROP(O,P,U)
                                 = (NULL(O) YIELDS U, CAR(O) = P YIELDS CDR(O),
                                      T YIELDS PROP(CDR(O),P,U))

07632  0634 00 4 07706   PROP  SXA     SAST1,4              SAVE LINK IR
07633  0131 00 0 00000         XCA                          PROPERTY TO AC
07634  0622 00 0 07645         STD SASP1                    SET TXH
07635  0402 00 0 00126         SUB SASQ1
07636  0622 00 0 07644         STD SASP2                    SET TXL
07637  0131 00 0 00000         XCA                          OBJECT TO AC
07640 -0734 00 4 00000   SASL1 PDX 0,4                      L = CDR(L)
                                INSERT TXH INSTRUCTION HERE IF NILL IS NADE NON-ZERO
07641 -3 00000 4 07651         TXL SASP3,4,0                NULL(L)
07642  0500 00 4 00000         CLA 0,4                      CWR(L)
07643  0734 00 4 00000         PAX 0,4                      CAR(L)
07644 -3 00000 4 07640   SASP2 TXL SASL1,4,**
07645  3 00000 4 07640   SASP1 TXH SASL1,4,**
07646 -0320 00 0 00133         ANA SASDM
07647  0534 00 4 07706         LXA     SAST1,4              RESTORE LINK IR
07650  0020 00 4 00001         TRA 1,4

07651 -0754 00 0 00000   SASP3 PXD     0,0                  CLEAR
07652 -0534 00 4 02531         LXD     $ARG3,4              INSPECT FUNCTIONAL ARGUMENT
07653  3 00012 4 07656         TXH     *+3,4,10             SKIP IF NOT A TXL
07654  0534 00 4 07706         LXA     SAST1,4
07655  0020 00 0 02531         TRA     $ARG3
07656  0600 00 0 02531         STZ     $ARG3
07657  0560 00 0 02531         LDQ     $ARG3
07660 -0754 00 4 00000         PXD     ,4
07661  0534 00 4 07706         LXA     SAST1,4              RESTORE LINK IR
07662  0020 00 0 13110         TRA     $APPLY

                             SASSOC(O,A,U)
                              = (NULL(A) YIELDS U, CAAR(A) YIELDS  CAR(A),
                                   T YIELDS SASSOC(O,CDR(A),U))

07663  0634 00 4 07706  SASSOC SXA     SAST1,4              SAVE LINK IR
07664  0634 00 2 07705         SXA     SAST2,2              SAVE IR 2
07665  0634 00 1 07703         SXA     SAST3,1              SAVE IR 1
07666  0622 00 0 07702         STD SASP7                    SET TXH
07667  0402 00 0 00126         SUB SASQ1
07670  0622 00 0 07701         STD SASP6                    SET TXL
07671  0131 00 0 00000         XCA                          PAIR LIST TO AC
07672 -0734 00 4 00000         PDX     0,4                  TO INDEX 4
07673 -3 00000 4 07710   SASP5 TXL SASP4,4,0                NULL(A)
                                INSERT TXH INSTRUCTION HERE IF NILL IS NADE NON-ZERO
07674  0500 00 4 00000         CLA 0,4                      CWR(A)
07675 -0734 00 4 00000         PDX ,4                       CDR(A)
```

```
07676  0734 00 2 00000          PAX ,2                    CAR(A)
07677  0500 00 2 00000          CLA ,2
07700  0734 00 1 00000          PAX     0,1               CAAR(A) TO INDX REGISTER
07701 -3 00000 1 07673  SASP6   TXL     SASP5,1,**        LOOK FOR ITEM
07702  3 00000 1 07673  SASP7   TXH     SASP5,1,**
07703  0774 00 1 00000  SAST3   AXT     **,1              FOUND ITEM, RESTORE IR 1
07704 -0754 00 2 00000          PXD     0,2               POINTER TO WORD
07705  0774 00 2 00000  SAST2   AXT     **,2              RESTORE IR 2
07706  0774 00 4 00000  SAST1   AXT     **,4              RESTORE LINK IR
07707  0020 00 4 00001          TRA 1,4

07710  0534 00 2 07705  SASP4   LXA     SAST2,2           RESTORE IR 2
07711  0534 00 1 07703          LXA     SAST3,1           RESTORE IR 1
07712  0020 00 0 07651          TRA     SASP3             EXECUTE SASSOC EXIT
                00126  SASQ1    SYN     $QD1
                00133  SASDM    SYN     $DMASK

07713  0100 00 4 00001  SPREAD  TZE     1,4               EXIT IF AGLIST IS NULL
07714  0634 00 4 07753          SXA     SPRX,4            SAVE LINK IR
07715 -0734 00 4 00000          PDX     0,4               POINTER TO ARG LIST
07716  0500 00 4 00000          CLA     0,4               FIRST WORD
07717  0560 00 0 15244          LDQ     =0                ZERO THE MQ
07720 -0765 00 0 00022          LGR     18                CAR TO CDR OF MQ
07721  0100 00 0 07752          TZE     NLY               GO IF A  SINGLE ARGUMENT
07722  0734 00 4 00000          PAX     0,4               POINTER TO NEXT WORD
07723  0500 00 4 00000          CLA     0,4               NEXT WORD
07724  0734 00 4 00000          PAX     0,4               POINTER TO ARGUMENT
07725 -0320 00 0 00133          ANA     $DMASK            MASK OUT ALL BUT DECREMENT
07726  0100 00 0 07751          TZE     TWA               GO IF 2 ARGUMENTS
07727 -0634 00 4 02530          SXD     $ARG2,4           PUT AWAY
07730  0634 00 2 07747          SXA     SPRY,2            SAVE INDEX 1 AND 2
07731  0634 00 1 07746          SXA     SPRZ,1
07732  0774 00 1 00022          AXT     18,1              20 IS MAX NO OF ARGS
07733 -0734 00 4 00000          PDX     0,4               REST OF ARG LIST TO IR 4
07734 -3 00000 4 07746  SPP1    TXL     SPRZ,4,0          GO IF END OF LIST
07735  0500 00 4 00000          CLA ,4
07736 -0734 00 4 00000          PDX ,4
07737  0734 00 2 00000          PAX ,2
07740 -0754 00 2 00000          PXD ,2
07741  0601 00 1 02553          STO     $ARG20+1,1
07742  2 00001 1 07734          TIX SPP1,1,1
07743                           ERROR   TMNARG            TOO MANY ARGUMENTS TO SPREAD

07746  0774 00 1 00000  SPRZ    AXT     **,1              RESTORE IR 1
07747  0774 00 2 00000  SPRY    AXT     **,2              DITTO IR 2
07750 -0534 00 4 02530          LXD     $ARG2,4           ARG 2
07751 -0754 00 4 00000  TWA     PXD     0,4               PUT IN DECREMENT AC
07752  0131 00 0 00000  NLY     XCA                       ARG 1 AND 2 TO RIGHT REGISTERS
07753  0774 00 4 00000  SPRX    AXT     **,4              RESTORE LINK IR
07754  0020 00 4 00001          TRA     1,4               EXIT
```

```
FUNCTION  ATTRIB(O,L)
ATTRIB(O,L)=/ CDR(O)=0 YIELDS (L REPLACES CDR(O))
                    ELSE ATTRIB(CDR(O),L)  /
```

LISP FUNCTIONS

```
                                R       HED
07755  0634 00 4 07767  ATTRIB  SXA  AT1,4
07756 -0100 00 0 07761           TNZ      ATRB                GO IF BEGINNING OF LIST
07757  0131 00 0 00000           XCA                          OTHERWISE EXIT WITH ARG 2
07760  0020 00 4 00001           TRA      1,4
07761 -0734 00 4 00000    ATRB   PDX  ,4                      0
07762  0500 00 4 00000           CLA  ,4
07763 -0320 00 0 00133           ANA  DMASK                   CDR(0)
07764 -0100 00 0 07761           TNZ  ATRB
07765  0131 00 0 00000           XCA                          ARG 2 TC AC
07766  0622 00 4 00000           STD  ,4
07767  0774 00 4 00000    AT1    AXT  **,4
07770  0020 00 4 00001           TRA  1,4
                   00133   DMASK  SYN      $DMASK


                                THE RPLACX FUNCTIONS  REPLACE THE X PART OF THE FIRST ARG
                                WITH THE SECOND ARGUMENT
                                THE VALUE OF RPLACA,RPLACD, AND RPLACW IS ZERO

                                S       HED
07771  0634 00 4 07776  RPLACA  SXA  REPL,4
07772 -0734 00 4 00000           PDX  0,4
07773 -0763 00 0 00022           LGL  18
07774  0621 00 4 00000           STA  0,4
07775 -0754 00 4 00000    RPLEX  PXD      0,4                  ARG1 TO AC AS ANSWER
07776  0774 00 4 00000    REPL   AXT      **,4                 RESTORE LINK IR
07777  0020 00 4 00001           TRA  1,4
10000  0634 00 4 07776  RPLACD  SXA  REPL,4
10001 -0734 00 4 00000           PDX  0,4
10002 -0620 00 4 00000           SLQ  0,4
10003  0020 00 0 07775           TRA      RPLEX               EXIT
10004  0634 00 4 07776  RPLACW  SXA  REPL,4
10005 -0734 00 4 00000           PDX  0,4
10006 -0600 00 4 00000           STQ  0,4
10007  0020 00 0 07775           TRA      RPLEX               EXIT


                                OBJECT GENERATOR

10010  0634 00 4 10030  GENSYM  SXA      GENX,4              SAVE LINK IR
10011  0500 00 0 10032           CLA      DIGIT               GET DIGITS
10012  0114 06 0 04315           CVR      BCDAD1,,6           ADD 1 IN BCD
10013  0601 00 0 10032           STO      DIGIT               UPDATE CELL
10014 -0501 00 0 15277           ORA      =HG00000
10015  0074 00 4 02674           TSX  $CONSW,4
10016  0560 00 0 15244           LDQ  =0
10017  0074 00 4 02714           TSX  $CONS,4
10020  0560 00 0 15244           LDQ  =0
10021  0074 00 4 02714           TSX  $CONS,4
10022  0131 00 0 00000           XCA
10023  0500 00 0 00151           CLA  GENPN
10024  0074 00 4 02714           TSX  $CONS,4
10025  0131 00 0 00000           XCA
10026  0500 00 0 00133           CLA      GENC
10027  0074 00 4 02714           TSX      $CONS,4
10030  0774 00 4 00000    GENX   AXT      **,4                RESTORE LINK IR
```

LISP FUNCTIONS

```
10031  0020 00 4 00001          TRA 1,4
                 00151   GENPN  SYN     PNAMED
                 00133   GENC   SYN     $DMASK
10032  000000000000     DIGIT  BCI     1,000000
10033 -0754 00 0 00000  OCTALP ZAC              CONVERT LEFT HALF OF MQ TO OCTAL
10034 -0763 00 0 00003          LGL     3            CLEAR AC AND DO SHIFT DANCE
10035  0767 00 0 00003          ALS     3
10036 -0763 00 0 00003          LGL     3
10037  0767 00 0 00003          ALS     3
10040 -0763 00 0 00003          LGL     3
10041  0767 00 0 00003          ALS     3
10042 -0763 00 0 00003          LGL     3
10043  0767 00 0 00003          ALS     3
10044 -0763 00 0 00003          LGL     3
10045  0767 00 0 00003          ALS     3
10046 -0763 00 0 00003          LGL     3
10047  0020 00 4 00001          TRA     1,4          EXIT
```

```
                                        INSERT  $EVQTZF
                                *       EVALQ   A SUCCESSOR TO THE APPLY OPERATOR, THE GRAND NEW
                                *               (AS OF 1 MARCH 1961) THE EVALQUOTE OPERATOR.
                                *
                                *  CHANGE MADE BY MOSES
10050  0074 00 4 01155  EVALQ   TSX     $SEEK,4
10051  0600 00 0 10321  EVALCH  STZ     EVQRTS          INITIALIZE TEST CELLS
10052  0600 00 0 02670          STZ     EVQB                DITTO
10053  0600 00 0 02667          STZ     EVQLS           CLEAR LIST USED TO REPLACE EVALQ BUFFER
10054  0074 00 4 04755  EVQRD    TSX     $READ,4         READ THE INPUT LISTS
10055  0340 00 0 10322          CAS     EVQSP               COMPARE WITH STOP ATOM
10056  0020 00 0 10060          TRA     *+2                 IS NOT
10057  0020 00 0 10102  EVQRF    TRA     EVQPO
10060  0340 00 0 10323          CAS     EVQEOF          MIGHT BE END OF INPUT FILE
10061  0020 00 0 10063          TRA     *+2             *ISN'T
10062  0020 00 0 10057          TRA     EVQRF           *YES.  CALL IT STOP.
10063  0520 00 0 02670          ZET     EVQB                SKIP IF FIRST LIST OF DOUBLET
10064  0020 00 0 10070          TRA     EVQA                IS SECOND LIST
10065 -0625 00 0 02670          STL     EVQB                FLIP SWITCH
10066  0601 00 0 10144          STO     EVQFN           SAVE FIRST LIST OF DOUBLET
10067  0020 00 0 10054          TRA     EVQRD               GET NEXT LIST
10070  0560 00 0 10144  EVQA     LDQ     EVQFN
10071  0074 00 4 02714          TSX     $CONS,4
10072  0560 00 0 15244          LDQ     =0
10073  0074 00 4 02714          TSX     $CONS,4
10074  0131 00 0 00000          XCA
10075  0500 00 0 02667          CLA     EVQLS
10076  0074 00 4 07426          TSX     $APPEND,4
10077  0601 00 0 02667          STO     EVQLS
10100  0600 00 0 02670          STZ     EVQB                FLIP SWITCH
10101  0020 00 0 10054          TRA     EVQRD
10102  0074 00 4 05452  EVQPO    TSX     TEREAD,4
                        * DELETED BY THE MAN HIMSELF
10103 -0625 00 0 10321          STL     EVQRTS              SET ERROR RETURN SWITCH
10104 -0534 00 4 02667  EVQS     LXD     EVQLS,4
10105  0500 00 4 00000          CLA     0,4
10106  0622 00 0 02667          STD     EVQLS
10107  0734 00 4 00000          PAX     0,4
10110  0500 00 4 00000          CLA     0,4
10111  0600 00 0 02526          STZ     $ALIST          JUST IN CASE ITS REALLY USEFUL
10112 -0734 00 4 00000          PDX     0,4                 MAKE AN ATOM TEST
10113  0560 00 0 15244          LDQ     =0
10114 -0765 00 0 00022          LGR     18                  SECOND LIST INTO MQ
10115 -0754 00 4 00000          PXD     0,4                 FIRST LIST INTO AC
                        *  HEADING KILLED BY THE PROPHET
10116 -0774 00 4 10132          AXC     EVQFT,4
10117  0634 00 4 10131          SXA     EVQD,4
10120  0774 00 4 13110          AXT     $APPLY,4            SET CELL OF PROGRAM TO BE EXECUTED
10121  0634 00 4 10132          SXA     EVQFT,4             INITIALIZE PROGRAM TO BE EXECUTED CELL
10122  0601 00 0 10316          STO     EVQAC               SAVE AC
10123 -0734 00 4 00000          PDX     0,4                 FIRST LIST TO IR 4
10124  0500 00 4 00000          CLA     0,4
10125  0734 00 4 00000          PAX     0,4
10126  3 77776 4 10244          TXH     EVQAT,4,-2          TRANSFER IF FIRST LIST IS ATOMIC
10127  0500 00 0 10316  EVQNF    CLA     EVQAC               RESTORE AC
```

```
10130  0600 00 0 02531  EVQZ   STZ    $ARG3              NULL ALIST FOR APPLY
10131  0774 00 4 00000   EVQD  AXT    **,4               RETURN INDEX REGISTER
10132  0020 00 0 00000   EVQFT TRA    **                 PROGRAM TO BE EXECUTED
10133  0601 00 0 02666         STO    EVQAN              SAVE ANSWER
10134                           TSSX   WRFLX,4            PRINT END OF EVALQUOTE MESSAGE
10135  0 00001 0 15312          =H VALUE,,1    ..
10136  0500 00 0 02666         CLA    EVQAN              PICK UP ANSWER
10137  0074 00 4 03564         TSX    $PRINT,4           PRINT IT
10140  0600 00 0 02666         STZ    EVQAN              ZERO TEMP STORAGE
10141 -0534 00 4 02667  EVQER  LXD    EVQLS,4
10142 -3 00000 4 10145         TXL    EVQDN,4,0
10143  0020 00 0 10104         TRA    EVQS
10144  0 00000 0 00000  EVQFN  PZE                      CONSTANT,MAY HAVE TO BE MARKED BY GC****
                        *
                        * EXIT FROM EVALQUOTE VIA CHNCOM
                        *
10145                   EVQDN  TSSX   CLOSE,4           CLOSE ALL FILES
10146  3 00000 0 15276         PTH    =HALL              CLOSE ALL FILES
10147  0500 00 0 15245         CLA    =1
10150                           TSSX   CHNCOM,4
10151  3 00000 0 00001         PTH    1                  MAY CALL DORMANT
10152  0020 00 0 10154         TRA    *+2
10153  0600 00 0 01056  EVALQT STZ    $OUTSW
10154                           TSSX   GETCOM,4
10155  0000 00 0 00001         HTR    1
10156 -0340 00 0 15302         LAS    =HNOCOMT
10157  0020 00 0 10161         TRA    *+2
10160  0020 00 0 10167         TRA    EVQT1
10161 -0340 00 0 00140         LAS    SEVENS
10162  0020 00 0 10173         TRA    NEWFIL             FILE NAME VIA TY
10163  0020 00 0 10165         TRA    *+2
10164  0020 00 0 10173         TRA    NEWFIL
10165                           TSSX   WRFLX,4
10166  0 00004 0 10201                LISTCM,,4          WRITE COMMENT
10167 -0500 00 0 15307  EVQT1  CAL    =H  LISP
10170  0602 00 0 01104         SLW    OUTFIL
10171  0074 00 4 05452         TSX    TEREAD,4           FOR GOOD MEASURE
10172  0020 00 0 00100         TRA    LISTNW             GO TO LISTEN PROGRAM INSTEAD OF LISP LISP
10173 -0340 00 0 15310  NEWFIL LAS    =H    $$           IS IT $$ SPECIAL CALL
10174  0020 00 0 10176         TRA    *+2                NO
10175  0020 00 0 10205         TRA    COMM               YES
10176  0602 00 0 01104         SLW    OUTFIL
10177  0602 00 0 01102         SLW    FILNAM
10200  0020 00 0 10050         TRA    EVALQ              SAME TO YOU.

10201  604331633047     LISTCM BCI    4, LITHP IS LITHTENING

10205  0774 00 2 00014  COMM   AXT    12,2
10206  0754 00 2 00000  COMLP  PXA    0,2                LOOP TO READ COMMAND INTO READ BUFFER
10207 -0760 00 0 00003         SSM
10210  0400 00 0 15256         ADD    =14                WANTS COMMAND WORDS 2-13
10211  0621 00 0 10213         STA    GETWD
10212                           TSSX   GETCOM,4
10213  0000 00 0 00000  GETWD HTR    **
10214  0602 00 2 06377         SLW    I$CELL,2
```

```
10215  2 00001 2 10206          TIX     COMLP,2,1
10216  0600 00 0 06402          STZ     I$RDLST         START FOOLING THE READ PROGRAM
10217  0600 00 0 05046          STZ     I$LRCIS         MAKE SURE IT DOESNT READ TY
10220  0500 00 0 15255          CLA     =12
10221  0621 00 0 05066          STA     I$RDWDS         TELL IT TO READ 12 WORDS
10222  0500 00 0 15251          CLA     =6
10223  0621 00 0 05065          STA     I$RDPTS         READ FIRST CHARACTER
10224  0621 00 0 05410          STA     I$PUTMC         STORE FIRST CHARACTER
10225 -0625 00 0 05276          STL     PRDIND          SAY PREVIOUS DEVICE WAS TY
10226 -0625 00 0 01215          STL     RDINDC          CURRENTLY READING TY
10227  0020 00 0 10051          TRA     EVALCH
                        *
                        * ERROR RETURNS CONTROL HERE
                        *
10230  0074 00 4 05452  EVQERR  TSX     TEREAD,4        CLEAN UP READ BUFFER
10231  0074 00 4 04230          TSX     TERPRI,4        CLEAN UP PRINT BUFFER
10232  0074 00 4 04434          TSX     TERPUN,4        CLEAN UP PUNCH BUFFER
10233  0074 00 4 01777          TSX     TERPDL,4        RESET PUSH DOWN LIST
10234  0520 00 0 10321          ZET     EVQRTS          SKIP IF IN READ IN SECTION OF EVALQUOT
10235  0020 00 0 10141          TRA     EVQER           EXECUTE NEXT DOUBLET
10236 -0625 00 0 10321          STL     EVQRTS          MOVE TO OPERATE SECTION OF EVALQUOTE
10237                           TSSX    WRFLX,4         MESSAGE THAT READ WAS ERROR TERMINATED
10240  0 00012 0 10324                  EVQRE,,10
10241  0500 00 0 02666          CLA     EVQAN           PICK UP LAST LIST READ IN
10242  0074 00 4 03564          TSX     $PRINT,4
10243  0020 00 0 10102          TRA     EVQPO
                        *
                        * CASE FOR ATOMIC FIRST LIST OF DOUBLET
                        *
10244 -0734 00 4 00000   EVQAT  PDX     0,4
10245 -3 00000 4 10127          TXL     EVQNF,4,0       EXIT IF END OF ATOM
10246  0500 00 4 00000          CLA     0,4             NEXT WORD
10247  0734 00 4 00000          PAX     0,4             CAR OF ATOM
10250 -0625 00 0 10320          STL     EVQST           SET SWITCH FOR SUBR OR EXPR
10251 -3 10464 4 10253          TXL     *+2,4,$SUBR-1   LOOK FOR $SUBR
10252 -3 10465 4 10304          TXL     EVQFS,4,$SUBR   TREAT AS FSUBR (ALMOST)
10253 -3 12246 4 10255          TXL     *+2,4,$EXPR-1   LOOK FOR $EXPR
10254 -3 12247 4 10262          TXL     EVQFX,4,$EXPR   TREAT AS FEXPR (ALMOST)
10255  0600 00 0 10320          STZ     EVQST           SET SWITCH FOR FSUBR OR FEXPR
10256 -3 12064 4 10260          TXL     *+2,4,$FSUBR-1  LOOK FOR FSUBR
10257 -3 12065 4 10304          TXL     EVQFS,4,$FSUBR
10260 -3 12227 4 10244          TXL     EVQAT,4,$FEXPR-1 LOOK FOR FEXPR
10261  3 12230 4 10244          TXH     EVQAT,4,$FEXPR
10262 -0734 00 4 00000   EVQFX  PDX     0,4             FOUND AN FEXPR
10263  0500 00 4 00000          CLA     0,4
10264  0734 00 4 00000          PAX     0,4             THE EXPRESSION FOR THE FEXPR
10265 -0754 00 4 00000          PXD     0,4             EXPRESSION TO AC
10266  0520 00 0 10320          ZET     EVQST           SKIP IF FEXPR
10267  0020 00 0 10130          TRA     EVQZ            GO TO APPLY CALL FOR EXPR
10270  0601 00 0 02666          STQ     EVQAN           SAVE THE EXPRESSION
10271 -0600 00 0 10317          STQ     EVQMQ           SAVE MQ
10272 -0754 00 0 00000          PXD     0,0             CLEAR
10273  0131 00 0 00000          XCA                     MQ AND
10274 -0754 00 0 00000          PXD     0,0             AC
10275  0074 00 4 02714          TSX     $CONS,4         NULL A LIST
```

```
10276  0131 00 0 00000          XCA                        INTO MQ
10277  0500 00 0 10317          CLA      EVQMQ             PUT SECOND LIST IN AC
10300  0074 00 4 02714          TSX      $CONS,4           CONS(L,A)
10301  0131 00 0 00000          XCA                        ANSWER TO ARG 2
10302  0500 00 0 02666          CLA      EVQAN             FEXPR
10303  0020 00 0 10130          TRA      EVQZ              GO TO APPLY FOR FEXPR
                        *
10304 -0734 00 4 00000   EVQFS  PDX      0,4               FOUND FSUBR, GET TXL INSTRUCTION
10305  0500 00 4 00000          CLA      0,4
10306  0734 00 4 00000          PAX      0,4
10307  0500 00 4 00000          CLA      0,4
10310  0621 00 0 10132          STA      EVQFT             SAVE ADDRESS
10311 -0754 00 0 00000          PXD      0,0               ZERO
10312  0131 00 0 00000          XCA                        THE MQ AND PUT LIST IN AC
10313  0520 00 0 10320          ZET      EVQST             SKIP IF FSUBR
10314  0074 00 4 07713          TSX      SPREAD,4          SPREAD THE ARGUMENTS
10315  0020 00 0 10131          TRA      EVQD              EXECUTE THE SUBR OR FSUBR
                        *
10316  0 00000 0 00000   EVQAC                             TEMPORARY STORAGE
10317  0 00000 0 00000   EVQMQ                             DITTO
10320  0 00000 0 00000   EVQST                             TEST CELL IS NON-ZERO FOR SUBR OR EXPR
10321  0 00000 0 00000  EVQRTS                            TEST CEEL IS ZERO DURING READ IN
10322  0 10504 0 00000   EVQSP           ,,$STOP           STOP ATOM
10323  0 07654 0 00000  EVQEOF          ,,$EOF
                        ****CHANGE*******
10324  512521243145      EVQRE BCI      9,READING TERMINATED BY AN ERROR. LAST LIST READ IN IS
10335  603333333333             BCI      1, .....
                        *
```

```
                                    INSERT  $CARXZF
                              H     HED
                              *     ERRORSET(E,N,SW,AL)
                              *
                              *     ERRORSET ATTEMPTS TO EVALUATE ITS FIRST ARGUMENT.  IF AN
                              *     ERROR OCCURS DURING THE EVALUATION, OR IF MORE THAN N CONS-S
                              *     OCCUR DURING THE EVALUATION, ERRORSET RETURNS WITH A VALUE OF F
                              *     AFTER RESTORING CONDITIONS TO WHAT THEY WERE BEFORE THE
                              *     ATTEMPTED EVALUATION.  IF THE EVALUATION SUCCEEDS, ERRORSET
                              *     RETURNS LIST OF THE RESULT.  IF SW = F, ERROR DIAGNOSTICS ARE
                              *     SUPPRESSED, AND IF SW = T, THEY ARE INCLUDED.  AL IS THE A-LIST
                              *     USED FOR THE EVALUATION.
                              *
10336 -0634 00 4 10440  ERRSET SXD     HORN,4
10337  0074 00 4 01723          TSX     $SAVE,4
10340 -3 10451 0 01752          TXL     $END8,,HORN+9
10341 -0634 00 2 10441          SXD     HORN+1,2
10342  0634 00 1 10441          SXA     HORN+1,1
10343  0604 00 0 10444          STI     HORN+4
10344 -0734 00 1 00000          PDX     0,1             EXPRESSION TO BE EVALUATED
10345 -0534 00 4 02531          LXD     $ARG3,4         ERROR BYPASS SWITCH
10346  0634 00 4 10442          SXA     ERNULL,4
10347  0131 00 0 00000          XCA
10350 -0734 00 2 00000          PDX     0,2             GET CONS COUNTER LIMIT
10351  0074 00 4 11423          TSX     FIXVAL,4
10352  0601 00 0 10446          STO     HORN+6
10353  0500 00 0 02726          CLA     $CNTR1          GET CURRENT CONS COUNT
10354 -0320 00 0 00132          ANA     $AMASK
10355  0400 00 0 03073          ADD     $CNTS
10356  0402 00 0 10446          SUB     HORN+6          COMPARE WITH THE LIMIT
10357 -0120 00 0 10367          TMI     OBOE            TRA IF COUNTER NEED NOT BE CHANGED
10360 -0760 00 0 00003          SSM                     NEG. NUMBER FOR GARBAGE COLLECTOR
10361  0601 00 0 10445          STO     HORN+5          SAVE (LIMIT - OLD COUNT)
10362  0500 00 0 10446          CLA     HORN+6          SET CONS COUNTER TO LIMIT
10363  0621 00 0 02726          STA     $CNTR1
10364 -0320 00 0 00135          ANA     PDTMSK
10365  0601 00 0 03073          STO     $CNTS
10366  0020 00 0 10370          TRA     *+2
10367  0600 00 0 10445   OBOE   STZ     HORN+5          TAKE LIMIT = OLD COUNT
10370  0560 00 0 02532          LDQ     $ARG4           A-LIST FOR EVALUATION
10371  0502 00 0 01727          CLS     $CPPI           SAVE PUSHDOWN POINTER
10372  0601 00 0 10446          STO     HORN+6
10373 -0625 00 0 10447          STL     TCOUNT          TURN ON CONS COUNTER
10374  0774 00 4 10403          AXT     BSOON,4         SET UP EXIT IN ERROR
10375  0634 00 4 10443          SXA     EREXIT,4
                              *     ATTEMPT TO PERFORM THE EVALUATION
10376 -0754 00 1 00000          PXD     0,1             EXPRESSION TO BE EVALUATED
10377  0074 00 4 13703          TSX     $EVAL,4
                              *     WE GET HERE IF THE EVALUATION WORKED
10400  0560 00 0 15244          LDQ     =0              FORM LIST OF THE RESULT
10401  0074 00 4 02714          TSX     $CONS,4
                              *     AN ERROR IN THIS CONS ACTS LIKE AN ERROR IN THE EVALUATION
10402  0020 00 0 10421          TRA     SHAWM           RESTORE PARAMETERS AND EXIT
                              *     WE GET HERE IN CASE OF ERROR
10403 -0535 00 4 10446   BSOON LDC     HORN+6,4        UNSAVE ALL RECURSIVE FUNCTIONS
```

```
10404 -0634 00 4 10417          SXD     TUBA,4          ENTERED SINCE THE ERROR
10405  0020 00 0 10416          TRA     TUBA-1
10406 -0534 00 4 01727  HARP    LXD     $CPPI,4
10407 -0500 00 4 77777          CAL     -1,4
10410 -0320 00 0 00134          ANA     $PMASK          TEST FOR STR FROM COMPILER
10411  0322 00 0 00130          ERA     $QP5
10412  0100 00 0 10415          TZE     *+3
10413  0074 00 4 01736          TSX     UNSAVE,4
10414  0020 00 0 10416          TRA     *+2
10415  0074 00 4 15027          TSX     C$UNWND,4
10416 -0535 00 4 01727          LDC     $CPPI,4
10417  3 00000 4 10406  TUBA    TXH     HARP,4,**
10420 -0754 00 0 00000          PXD     0,0             RETURN VALUE OF NIL
                        *       RESTORE PARAMETERS FOR EITHER KIND OF EXIT
10421  0601 00 0 10446  SHAWM   STO     HORN+6          SAVE EXIT VALUE
10422  0500 00 0 02726          CLA     $CNTR1          RESTORE CONS COUNTER
10423 -0320 00 0 00132          ANA     $AMASK
10424  0400 00 0 03073          ADD     $CNTS
10425  0402 00 0 10445          SUB     HORN+5
10426  0621 00 0 02726          STA     $CNTR1
10427 -0320 00 0 00135          ANA     PDTMSK
10430  0601 00 0 03073          STO     $CNTS
10431  0534 00 1 10441          LXA     HORN+1,1        RESTORE INDICATORS, IR1, AND IR2
10432 -0534 00 2 10441          LXD     HORN+1,2
10433  0441 00 0 10444          LDI     HORN+4
10434  0500 00 0 10446          CLA     HORN+6          PICK UP EXIT VALUE
10435  0074 00 4 01736          TSX     UNSAVE,4        RESTORE HORN BLOCK
10436 -0534 00 4 10440          LXD     HORN,4          RESTORE IR4 AND EXIT
10437  0020 00 4 00001          TRA     1,4
                        *       PROTECTED TEMPORARY STORAGE FOR ERRORSET
                                HEAD    H
10440 -0 00000 0 12354   HORN   MZE     ERSETO          (+0) ERRORSET OBJECT IN A, IR4 IN D
10441 -0 00000 0 00000          MZE                     (+1) IR1 IN A, IR2 IN D
10442 -0 00000 0 10442  ERNULL  MZE     *               (+2) ZERO MEANS SKIP DIAGNOSTICS
TD    10443 -3 00000 0 10230  EREXIT  TXL     EVQERR          (+3) EXIT INSTRUCTION FOR $ERROR
10444 -0 00000 0 00000          MZE                     (+4) INDICATORS
10445 -0 00000 0 00000          MZE                     (+5) CONS COUNTER INCREMENT
10446 -0 00000 0 00000  NUBPDL  MZE                     (+6) PDL BACKUP POINT IN D
10447 -0 00000 0 00000  TCOUNT  MZE                     (+7) NON-ZERO ACTIVATES CONS COUNTER
                                HEAD    0
                        10443   TERA2   SYN     EREXIT
```

```
                                *
                                *       HEAD    H
                                *
                                *       EXTENDED CAR S AND CDR S FOR THE INTERPRETER
                                *
10450  0634 00 4 10461  CAAARX  SXA     CAX,4           SAVE LINK IR
10451 -0734 00 4 00000           PDX     0,4
10452  0500 00 4 00000           CLA     0,4
10453  0734 00 4 00000           PAX     0,4
10454  0500 00 4 00000    AA     CLA     0,4
10455  0734 00 4 00000           PAX     0,4
10456  0500 00 4 00000    A      CLA     0,4
10457  0734 00 4 00000           PAX     0,4
10460 -0754 00 4 00000           PXD     0,4
10461  0774 00 4 00000    CAX    AXT     **,4            RESTORE LINK IR
10462  0020 00 4 00001           TRA     1,4             EXIT
                                *
10463  0634 00 4 10461  CAADRX  SXA     CAX,4
10464 -0734 00 4 00000           PDX     0,4
10465  0500 00 4 00000           CLA     0,4
10466 -0734 00 4 00000    AAX    PDX     0,4
10467  0020 00 0 10454           TRA     AA
                                *
10470  0634 00 4 10461  CADARX  SXA     CAX,4
10471 -0734 00 4 00000           PDX     0,4
10472  0500 00 4 00000           CLA     0,4
10473  0734 00 4 00000           PAX     0,4
10474  0500 00 4 00000    AD     CLA     0,4
10475 -0734 00 4 00000           PDX     0,4
10476  0020 00 0 10456           TRA     A
10477  0634 00 4 10461  CADDRX  SXA     CAX,4
10500 -0734 00 4 00000           PDX     0,4
10501  0500 00 4 00000           CLA     0,4
10502 -0734 00 4 00000    ADX    PDX     0,4
10503  0020 00 0 10474           TRA     AD
                                *
10504  0634 00 4 10461  CAARXX  SXA     CAX,4
10505  0020 00 0 10466           TRA     AAX
                                *
10506  0634 00 4 10461  CADRXX  SXA     CAX,4
10507  0020 00 0 10502           TRA     ADX
                                *
10510  0634 00 4 10520  CDAARX  SXA     CDX,4
10511 -0734 00 4 00000           PDX     0,4
10512  0500 00 4 00000           CLA     0,4
10513  0734 00 4 00000           PAX     0,4
10514  0500 00 4 00000    DA     CLA     0,4
10515  0734 00 4 00000           PAX     0,4
10516  0500 00 4 00000    D      CLA     0,4
10517 -0320 00 0 00133           ANA     $DMASK
10520  0774 00 4 00000    CDX    AXT     **,4
10521  0020 00 4 00001           TRA     1,4
10522  0634 00 4 10520  CDADRX  SXA     CDX,4
10523 -0734 00 4 00000           PDX     0,4
10524  0500 00 4 00000           CLA     0,4
```

MORE LISP FUNCTIONS

```
10525 -0734 00 4 00000   DAX    PDX     0,4
10526  0020 00 0 10514          TRA     DA
                        *
10527  0634 00 4 10520   CDDARX SXA     CDX,4
10530 -0734 00 4 00000          PDX     0,4
10531  0500 00 4 00000          CLA     0,4
10532  0734 00 4 00000          PAX     0,4
10533  0500 00 4 00000   DD     CLA     0,4
10534 -0734 00 4 00000          PDX     0,4
10535  0020 00 0 10516          TRA     D
                        *
10536  0634 00 4 10520   CDDDRX SXA     CDX,4
10537 -0734 00 4 00000          PDX     0,4
10540  0500 00 4 00000          CLA     0,4
10541 -0734 00 4 00000   DDX    PDX     0,4
10542  0020 00 0 10533          TRA     DD
                        *
10543  0634 00 4 10520   CDARXX SXA     CDX,4
10544  0020 00 0 10525          TRA     DAX
                        *
10545  0634 00 4 10520   CDDRXX SXA     CDX,4
10546  0020 00 0 10541          TRA     DDX
                        *
                                 HEAD    C
10547  0634 00 4 10561   GET    SXA     GETX,4          SAVE LINK IR
10550  0601 00 0 10564          STO     GETL
10551  0500 00 0 10563          CLA     FCN31
10552  0601 00 0 02531          STO     $ARG3
10553  0500 00 0 10564          CLA     GETL
10554  0074 00 4 07632          TSX     $PROP,4
10555 -0734 00 4 00000          PDX     0,4
10556  0500 00 4 00000          CLA     0,4
10557  0734 00 4 00000          PAX     0,4
10560 -0754 00 4 00000          PXD     0,4
10561  0774 00 4 00000   GETX   AXT     **,4            RESTORE  LINK IR
10562  0020 00 4 00001          TRA     1,4
10563 -3 00000 0 10561   FCN31  TXL     GETX,,0
10564  0 00000 0 00000   GETL
                        *
                        * COMPAT        FUNCTIONAL ARGUMENT LINKAGE PROGRAM BETWEEN COMPILED
                        *               PROGRAMS AND APPLY FOR S-EXPRESSION FUNCTIONAL ARGUMENTS
                        *
10565  0634 00 4 10604   COMPAT SXA     CX,4            SAVE INDEX REGISTERS
10566  0634 00 2 10605          SXA     CY,2
10567  0601 00 0 02527          STO     $ARG1           SAVE AC
10570 -0600 00 0 02530          STQ     $ARG2           DITTO MQ
10571  0560 00 0 15244          LDQ     =0              END OF ARGUMENT LIST
10572  0500 00 4 00001          CLA     1,4             ARGUMENTS FOR COMPAT
10573  0622 00 0 10607          STD     CA              S-EXPRESSION FUNCTIONAL ARGUMENT
10574  0737 00 2 00000          PAC     0,2             COMPLEMENT NUMBER OF ARGUMENTS
10575 -3 00000 2 10602   CL     TXL     CD,2,0          GO WHEN ALL DONE
10576  0500 00 2 02526          CLA     $ARG1-1,2       PICK UP ARGUMENT
10577  0074 00 4 02714          TSX     $CONS,4         CONS ON TO ARGUMENT LIST
10600  0131 00 0 00000          XCA                     LIST TO MQ
10601  1 00001 2 10575          TXI     CL,2,1          GO BACK FOR NEXT
```

```
10602  0500 00 0 10607   CD     CLA     CA                FUNCTIONAL ARGUMENT
10603  0600 00 0 02531          STZ     $ARG3             ZERO PAIR LIST
10604  0774 00 4 00000   CX     AXT     **,4              RESTORE INDEX REGISTERS
10605  0774 00 2 00000   CY     AXT     **,2
10606  1 77777 4 13110          TXI     $APPLY,4,-1       GO TO APPLY AND ADJUST EXIT INDEX
10607  0 00000 0 00000   CA             HED               S-EXPRESSION GOES HERE
                         F      HED
                         *      PACK(CHAR)
                         *
                         *      PACK ADDS ANOTHER CHARACTER TO THE CHARACTER BUFFER BOFFO
                         *
                         *
10610  0771 00 0 00022   PACK   ARS     18                GET CHARACTER CODE FROM
10611  0402 00 0 00166          SUB     HORG                LOCATION OF OBJECT
10612 -0765 00 0 00006          LGR     6                 PUT NEW CHARACTER INTO PACKED WORD
10613 -0500 00 0 11130          CAL     CHARS
10614  0140 00 0 10615          TOV     *+1               SHUT OFF OVERFLOW LIGHT
10615 -0763 00 0 00006          LGL     6
10616  0140 00 0 10622          TOV     B5                IF WORD FULL, PUT IT IN BUFFER
10617  0602 00 0 11130          SLW     CHARS
10620 -0754 00 0 00000          PXD     ,0                CLEAR AC FOR EXIT
10621  0020 00 4 00001          TRA     1,4               EXIT
10622  0634 00 4 10631   B5     SXA     B1,4              SAVE IR4
10623  0774 00 4 00024   BFLOC  AXT     20,4              ADDRESS HAS INDEX FOR BOFFO
10624  0602 00 4 11156          SLW     BOFFO,4           STORE FULL WORD OF CHARACTERS
10625 -2 00001 4 10634          TNX     B3,4,1            IF BUFFER FULL, TRANSFER
10626  0500 00 0 15245          CLA     =1                WHEN 1 SHIFTS PAST P BIT,
10627  0601 00 0 11130          STO     CHARS               NEW WORD HAS 6 CHARACTERS
10630  0634 00 4 10623          SXA     BFLOC,4           SAVE BUFFER INDEX
10631  0774 00 4 00000   B1     AXT     ,4                RESTORE IR4
10632 -0754 00 0 00000          PXD     ,0                CLEAR AC FOR EXIT
10633  0020 00 4 00001          TRA     1,4               EXIT
10634 -3 00000 4 10637   B3     TXL     B4,4,0            IF MORE THAN 120 CHARS, TRANSFER
10635  0634 00 0 10623          SXA     BFLOC,0           SET INDEX TO SHOW BUFFER FILLED
10636  0020 00 0 10644          TRA     B6
10637  0074 00 4 10721   B4     TSX     $MKNAM,4          FORM OBJECT FOR ERROR PRINTOUT
10640  0074 00 4 06406          TSX     INTRN1,4
10641                    ERROR  PKG120          TOO MANY CHARACTERS IN BOFFO
10644  0500 00 0 00140   B6     CLA     SEVENS            BIT 1 IN CHARS WILL MAKE
10645  0601 00 0 11130          STO     CHARS               WORD LOOK FULL
10646  0020 00 0 10631          TRA     B1

                         *      NUMOB
                         *
                         *      NUMOB MAKES A NUMERICAL OBJECT CORRESPONDING TO THE BCD
                         *      CHARACTERS IN THE BUFFER BOFFO.
                         *
                         *      THIS ROUTINE HAS CROSS-REFERENCES TO THE INNARDS OF NUMBR
                         *
10647  0634 00 4 10712   NUMOB  SXA     GV1,4             SAVE IR4
10650  0140 00 0 10651          TOV     *+1               SHUT OFF OVERFLOW LIGHT
```

```
10651 -0500 00 0 11130          CAL     CHARS           SHIFT SEVENS INTO LAST PACKED WORD
10652  0560 00 0 00140          LDQ     SEVENS
10653 -0763 00 0 00006          LGL     6
10654 -0140 00 0 10653          TNO     *-1             DONE WHEN 1 PASSES THROUGH P BIT
10655  0534 00 4 10623          LXA     BFLOC,4         PUT LAST WORD INTO BOFFO
10656  0602 00 4 11156          SLW     BOFFO,4
10657  0500 00 0 11120          CLA     PARAM           INPUT PARAMETER FOR NUMBR IS
10660  0074 00 4 06607          TSX     NUMBR,4           BEGINNING OF BOFFO
10661  0100 00 0 10714          TZE     GV3             ERROR IF ZERO IN AC
10662  0120 00 0 10670          TPL     GV2             TRANSFER IF FIXED POINT OUTPUT
10663  0131 00 0 00000          XCA                     GET NUMBER FROM MQ
10664  0560 00 0 00143          LDQ     FLOS            FLOATING POINT SIGNAL
10665  0074 00 4 11163          TSX     $MKNO,4         FORM OBJECT
10666  0534 00 4 10712          LXA     GV1,4           RESTORE IR4
10667  0020 00 0 10753          TRA     CLEAR           RESET BOFFO AND EXIT
10670 -0760 00 0 00001  GV2     PBT                     OCTAL SIGNAL IN NUMBR OUTPUT
10671  0020 00 0 10677          TRA     GV6             TRA IF NOT OCTAL
10672  0131 00 0 00000          XCA
10673  0560 00 0 00150          LDQ     $OCTD           MAKE OCTAL NUMBER
10674  0074 00 4 11163          TSX     $MKNO,4
10675  0534 00 4 10712          LXA     GV1,4
10676  0020 00 0 10753          TRA     CLEAR           RESET BOFFO AND EXIT
10677  0131 00 0 00000  GV6     XCA                     BRING THE NUMBER TO THE AC
10700 -0120 00 0 10710          TMI     GV4             TEST FOR DIGITS 0 THRU 9
10701  0340 00 0 00074          CAS     $Q10
10702  0020 00 0 10710          TRA     GV4
10703  0020 00 0 10710          TRA     GV4
10704  0361 00 0 00166          ACL     HORG                    FORM OBJECT DIRECTLY
10705  0767 00 0 00022          ALS     18
10706  0534 00 4 10712          LXA     GV1,4                   RESTORE IR4
10707  0020 00 0 10753          TRA     CLEAR
10710  0560 00 0 00142  GV4     LDQ     FIXS                    FIXED POINT SIGNAL FOR $MKNO
10711  0074 00 4 11163          TSX     $MKNO,4                 FORM NUMERICAL OBJECT
10712  0774 00 4 00000  GV1     AXT     ,4                      RESTORE IR4
10713  0020 00 0 10753          TRA     CLEAR                   RESET BOFFO AND EXIT
10714                    GV3     ERROR   NUMOB           FLOATING POINT NUMBER OUT OF RANGE
10717 -0754 00 0 00000          PXD     0,0                     RETURN NIL
10720  0020 00 0 10712          TRA     GV1
                        *
                        *       THIS ROUTINE USES $CONS, $MKNO,$ZERO,$ERROR, AND $ERORI
```

```
                        *       MKNAM AND CLEARBUFF
                        *
                        *
                        *       CLEARBUFF STARTS AT CLEAR AND RESETS THE BUFFER BOFFO TO
                        *       THE BEGINNING
                        *
                        *       MKNAM() HAS AS OUTPUT A PNAME LIST STRUCTURE CORRESPONDING
                        *       TO THE CHARACTERS IN THE BUFFER BOFFO.  THE BEGINNING OF
                        *       BOFFO IS RESET.
                        *
```

```
                                *       THIS ROUTINE HAS CROSS-REFERENCES TO THE INNARDS OF PACK.
                                *
10721  0634 00 4 10761   MKNAM SXA     BB1,4          SAVE IR4
10722  0634 00 2 10751          SXA     BBIR2,2        SAVE IR2
10723 -0500 00 0 11130          CAL     CHARS          IF C(CHARS) = 1, CHARS CONTAINS
10724 -0340 00 0 15245          LAS     =1               NO SIGNIFICANT CHARACTERS
10725  0020 00 0 10730          TRA     BB5
10726 -0754 00 0 00000          PXD     ,0
10727  0020 00 0 10741          TRA     BB2              NO SIGNIFICANT CHARACTERS IN CHARS
10730  0140 00 0 10731   BB5    TOV     *+1            SHUT OFF OVERFLOW LIGHT
10731  0560 00 0 00140          LDQ     SEVNS          SHIFT SEVENS INTO LAST WORD
10732 -0763 00 0 00006          LGL     6                OF LIST
10733 -0140 00 0 10732          TNO     *-1
10734  0602 00 0 11121          SLW     T1             PUT P BIT INTO SIGN
10735  0500 00 0 11121          CLA     T1
10736  0074 00 4 02674          TSX     $CONSW,4       FORM POINTER TO LAST WORD OF LIST
10737  0560 00 0 15244          LDQ     =0
10740  0074 00 4 02714          TSX     $CONS,4
10741  0534 00 2 10623   BB2    LXA     BFLOC,2        LOC OF LAST SIGNIFICAN BUFFER WORD
10742  3 00023 2 10751   BB4    TXH     BBIR2,2,19     TRA IF BUFFER EXHAUSTED
10743  0602 00 0 02671          SLW     BBPNT          SAVE DECREMENT FOR FUTURE USE
10744  0500 00 2 11155          CLA     BOFFO-1,2      GET NEXT WORD OF BUFFER
10745  0074 00 4 02674          TSX     $CONSW,4       TACK IT ONTO FRONT OF LIST
10746  0560 00 0 02671          LDQ     BBPNT
10747  0074 00 4 02714          TSX     $CONS,4
10750  1 00001 2 10742          TXI     BB4,2,1        MOVE TO NEXT WORD OF BUFFER
10751  0774 00 2 00000   BBIR2  AXT     **,2           RESTORE IR2
10752  0020 00 0 10754          TRA     BB3            RESET POSITION IN BOFFO
10753  0634 00 4 10761   CLEAR  SXA     BB1,4          ENTRANCE FOR CLEARING BUFFER
10754  0560 00 0 15245   BB3    LDQ     =1             RESET CHARS CELL TO 0 CHARACTERS
10755 -0600 00 0 11130          STQ     CHARS
10756  0774 00 4 00024          AXT     20,4           SET INDEX IN PACK FOR FIRST
10757  0634 00 4 10623          SXA     BFLOC,4          BUFFER WORD
10760  0600 00 0 02671          STZ     BBPNT          AVOID UNNECESSARY GARBAGE COLL.
10761  0774 00 4 00000   BB1    AXT     ,4             RESTORE IR4
10762  0020 00 4 00001          TRA     1,4            EXIT
```

MORE LISP FUNCTIONS

```
                        INSERT  $ADVIFE



                *       ADVANCE, STARTREAD, AND ENDREAD PROGRAMS
                *
                *       ADVANCE SETS CURCHAR TO THE NEXT CHARACTER
                *       STARTREAD READS A NEW RECORD
                *       ENDREAD MOVES TO THE END OF THE CURRENT RECORD AND
                *         GIVES ERROR OUTPUT, IF ANY

10763 -0634 00 4 10772  ADVANC SXD PORK,4               SAVE IR4
10764 -0534 00 4 11035          LXD CHPOS,4             FIND NO. OF CHARS. LEFT IN PACKED
10765  2 00006 4 11005          TIX CHOPS,4,6             WORD
10766 -0534 00 4 11036          LXD WDNUM,4             FIND NEW PACKED WORD
10767  2 00001 4 11001          TIX LAMB,4,1            IF NEW RECORD NEEDED, CONTINUE
10770 -0520 00 0 11131          NZT EORTS               IF NONZERO GIVE EOR AS OUTPUT CHAR-
10771  0020 00 0 10774          TRA JOYCE                 ACTER, OTHERWISE READ NEW RECORD
10772  1 00000 0 11032  PORK    TXI SUZIE,,0            READ A NEW RECORD
10773 -0634 00 4 10772  STREAD  SXD PORK,4              SAVE IR4
10774                   JOYCE   TSSX    RDFLXA,4
10775  0 00014 0 11100                  BUFF-12,,12    READ TY DIRECTLY,NO EOF OR ERROR RETURNS
10776 -0625 00 0 11131          STL EORTS               SET SIGNAL FOR EOR OUTPUT NEXT TIME
10777  0600 00 0 11162          STZ $CHACT              INITIALIZE CHARACTER COUNT
11000  0774 00 4 00014          AXT 12,4                SET INDEX FOR START OF INPUT BUFFER
11001 -0634 00 4 11036  LAMB    SXD WDNUM,4
11002  0500 00 4 11114          CLA BUFF,4              PICK UP NEW PACKED WORD FROM
11003  0601 00 0 11117          STO PWORD                 INPUT BUFFER AND STORE IT
11004  0774 00 4 00044          AXT 36,4                INITIALIZE POSITION IN PACKED WORD
11005 -0634 00 4 11035  CHOPS   SXD CHPOS,4
11006 -0754 00 0 00000          PXD ,0                  PICK OFF ONE CHARACTER
11007  0560 00 0 11117          LDQ PWORD
11010 -0763 00 0 00006          LGL 6
11011 -0600 00 0 11117          STQ PWORD               SAVE SHIFTED PACKED WORD
11012  0734 00 4 00000          PAX 0,4
11013  3 00055 4 11016          TXH     SHANK,4,45
11014 -3 00054 4 11016          TXL     SHANK,4,44     TEST FOR CARRIAGE RETURN CHARACTER***
11015  0020 00 0 11030          TRA     ENDRED+1       YIELD EOR CHARACTER,READ TY NEXT***
11016  1 07642 4 11017  SHANK   TXI **+1,4,$HOO          POINTER TO NEW CHARACTER OBJECT
11017  0500 00 0 11162  BACON   CLA $CHACT              BUMP CHARACTER COUNT
11020  0400 00 0 15245          ADD =1
11021  0601 00 0 11162          STO $CHACT
11022 -0754 00 4 00000          PXD ,4                  SET CURCHAR TO NEW CHARACTER
11023  0602 00 0 11161          SLW $CURC               POINTER IN DECREMENT FOR BIN
11024  0634 00 4 11160          SXA $CURC1,4            POINTER IN ADDRESS FOR APVAL1
11025 -0534 00 4 10772          LXD PORK,4              RESTORE IR4
11026  0020 00 4 00001          TRA 1,4                 RETURN
11027 -0634 00 4 10772  ENDRED  SXD PORK,4              SAVE IR4 FOR EXIT (ENDREAD ENTRANCE)
11030 -0634 00 0 11035          SXD CHPOS,0             SET CHARACTER POSITION AND WORD
11031 -0634 00 0 11036          SXD WDNUM,0             NUMBER AT END OF RECORD
11032  0534 00 4 00170  SUZIE   LXA EOR,4               LOAD END OF RECORD CHARACTER
11033  0600 00 0 11131          STZ EORTS               SIGNAL TO READ NEW RECORD NEXT TIME
11034  0020 00 0 11017          TRA BACON
```

MORE LISP FUNCTIONS

```
                                     ****
                                     *
                            00140     SEVNS SYN       SEVENS
          11035  0 00000 0 00000    CHPOS  PZE        ,,**
          11036  0 00000 0 00000    WDNUM  PZE        ,,**
```

ALPHABETIC FUNCTIONS

```
                                                DIGIT(CHAR)
          11037   0340 00 0 00171   DIGIT  CAS HOL9
          11040   0020 00 0 11044          TRA AL5
          11041   0761 00 0 00000          NOP
          11042   0500 00 0 00126          CLA $QD1
          11043   0020 00 4 00001          TRA 1,4
          11044  -0754 00 0 00000   AL5    PXD ,0
          11045   0020 00 4 00001          TRA 1,4
```

```
                                     *      UNPACK(NAME)
                                     *
                                     *      UNPACK(NAME) GIVES A LIST OF THE CHARACTER OBJECTS
                                     *      IN THE CELL -NAME-, UP TO THE FIRST 77.
                                     *
          11046   0634 00 4 11075   UNPACK SXA       UPI4,4              SAVE IR2 AND IR4
          11047   0634 00 2 11076          SXA       UPI2,2
          11050  -0734 00 4 00000          PDX       ,4                  PUT ARGUMENT CELL IN MQ
          11051   0560 00 4 00000          LDQ       0,4
          11052   0774 00 2 00006          AXT       6,2
          11053  -0754 00 0 00000   UP2    PXD       ,0                  LOOK AT A CHARACTER
          11054  -0763 00 0 00006          LGL       6
          11055   0340 00 0 15271          CAS       =077
          11056   1 00001 2 11062          TXI       UP1,2,1             ADJUST IR2 FOR CHARACTER
          11057   1 00001 2 11062          TXI       UP1,2,1               COUNT
          11060   0601 00 2 11127          STO       T1+6,2              STORE THE CHARACTER
          11061   2 00001 2 11053          TIX       UP2,2,1

          11062   0600 00 0 02671   UP1    STZ       UPLST               SET END OF LIST TO NIL
          11063   3 00006 2 11073   UP4    TXH       UP3,2,6             EXIT IF ALL CHARACTERS LISTED
          11064   0500 00 2 11127          CLA       T1+6,2              PICK UP NEXT CHARACTER
          11065   0400 00 0 00166          ADD       HORG                  AND FORM OBJECT
          11066   0767 00 0 00022          ALS       18
          11067   0560 00 0 02671          LDQ       UPLST
          11070   0074 00 4 02714          TSX       $CONS,4             PUT CHAR AT HEAD OF LIST
```

```
11071  0601 00 0 02671          STO    UPLST
11072  1 00001 2 11063          TXI    UP4,2,1
11073  0500 00 0 02671   UP3    CLA    UPLST              RETURN WITH LOCATION OF LIST
11074  0600 00 0 02671          STZ    UPLST              AVOID UNNECESSARY GARBAGE CELL.
11075  0774 00 4 00000   UPI4   AXT    **,4               RESTORE IR4 AND IR2
11076  0774 00 2 00000   UPI2   AXT    **,2
11077  0020 00 4 00001          TRA    1,4                EXIT
                         *
                         *      THIS ROUTINE USES $CONS


                                * * * * * * * * * * * * *  * * * * * * * * * * * * * * * * 

                                          STORAGE

              00166     HORG  SYN $H00A
              00170     EOR   SYN $H72A
              00171     HOL9  SYN $H11D
              00174     HOL40 SYN $H40D
11114                   BUFF  BES      12             BUFFER FOR INPUT RECORD
11117                         BES      3              ROOM FOR EXTRA WORDS IN READ-IN
11117                  PWORD  BSS 1
11120  0 00001 0 11132  PARAM PZE      BOFFO-20,,1
              00143     FLOS  SYN      FLOATD
11121                   T1    BSS      7
              00142     FIXS  SYN      $FIXD
11130                   CHARS BSS      1
11131                   EORTS BSS 1                   NONZERO INDICATES EOR OUTPUT CHAR
11156                   BOFFO BES      20
11156                         BSS      1              JUNK WORD FOR BOFFO REMNANTS
              02671     UPLST SYN      BBPNT          CUMULATIVE LIST OF CHARACTERS
11157                         BSS      1
11160  0 00000 0 00000 CURC1  PZE                     POINTER APPEARS IN ADDRESS
11161  0 00000 0 00000 CURC   PZE                     POINTER APPEARS IN DECREMENT
11162  0 00000 0 00000  CHACT PZE                     CHARACTER COUNT
                       *
                       *   MKNO     A FUNCTION OF TWO ARGUMENTS, THE FIRST IS A NUMBER, THE SECO
                       *            ND IS A TYPE (FLO OR FIX). MKNO FORMS A NON'UNIQUE NUMBER
11163  0634 00 4 11205  MKNO  SXA      MKIR,4         SAVE LINK IR
11164 -0600 00 0 02672        STQ      MKT1           TYPE OF NUMBER TO MQ
11165  0074 00 4 02674        TSX      $CONSW,4
11166  0131 00 0 00000        XCA
11167  0500 00 0 00133        CLA      $DMASK
11170  0074 00 4 02714        TSX      $CONS,4
11171 -0534 00 4 02672        LXD      MKT1,4         TYPE TO IR 4
11172  0622 00 0 02672        STD      MKT1
11173  0500 00 0 00125        CLA      $QT5           ASSUME IT IS OCTAL
11174 -3 12134 4 11177        TXL      *+3,4,$FIX-1
11175  3 12135 4 11177        TXH      *+2,4,$FIX
11176  0500 00 0 00122        CLA      $QT1
11177 -3 12115 4 11202        TXL      *+3,4,$FLOAT-1
11200  3 12116 4 11202        TXH      *+2,4,$FLOAT
```

```
11201  0500 00 0 00123         CLA     $QT2
11202 -0534 00 4 02672         LXD     MKT1,4            LOCATION OF NUMBER
11203 -0602 00 4 00000         ORS     0,4               PUT IN NUMBER FLAG
11204 -0754 00 4 00000         PXD     0,4               ANSWER TO AC
11205  0774 00 4 00000   MKIR  AXT     **,4              RESTORE LINK IR
11206  0020 00 4 00001         TRA     1,4
                        *
                        *
                        H      HED
                        *      LOGOR, LOGAND, AND LOGXOR
                        *
                        *      THESE FUNCTIONS TAKE THE LOGICAL AND, LOGICAL OR, AND LOGICAL
                        *      EXCLUSIVE OR RESPECTIVELY OF THEIR ARGUMENTS, WHICH ARE NUMBER
                        *      OBJECTS.  THE RESULT IS AN OCTAL NUMBER OBJECT.
                        *
11207  0100 00 4 00001   LOGOR TZE     1,4               RETURN 0 IF 0 INPUT
11210 -0634 00 4 11303         SXD     T1,4              SAVE IR4
11211  0774 00 4 11625         AXT     $PJ36,4      LOGOR ATOM
11212  0634 00 4 11303         SXA     T1,4              SET FUNCTION ON PDL
11213  0074 00 4 01723         TSX     $SAVE,4
11214 -3 11305 0 01770         TXL     $END1,,T1+2       SAVE 1 ITEM
11215  0074 00 4 14225         TSX     $EVLIS,4          EVALUATE LIST OF ARGUMENTS
11216  0074 00 4 01736         TSX     UNSAVE,4
11217  0560 00 0 15244         LDQ     =0                OR OF NO ARGUMENTS
11220 -0600 00 0 11304         STQ     T1+1
11221  0560 00 0 11274         LDQ     ORS               INSTRUCTION FOR INNER LOOP
11222  0020 00 0 11252         TRA     LOG2
                        *
11223  0100 00 4 00001  LOGAND TZE     1,4               EXIT WITH 0 IF 0 INPUT
11224 -0634 00 4 11303         SXD     T1,4              SAVE IR4
11225  0774 00 4 11636         AXT     $PJ37,4      LOGAND ATOM
11226  0634 00 4 11303         SXA     T1,4              SET FUNCTION ON PDL
11227  0074 00 4 01723         TSX     $SAVE,4
11230 -3 11305 0 01770         TXL     $END1,,T1+2       SAVE 1 ITEM
11231  0074 00 4 14225         TSX     $EVLIS,4          EVALUATE LIST OF ARGUMENTS
11232  0074 00 4 01736         TSX     UNSAVE,4
11233  0560 00 0 00140         LDQ     SEVENS            AND OF NO ARGUMENTS
11234 -0600 00 0 11304         STQ     T1+1
11235  0560 00 0 11275         LDQ     ANS               INSTRUCTION FOR INNER LOOP
11236  0020 00 0 11252         TRA     LOG2
                        *
11237  0100 00 4 00001  LOGXOR TZE     1,4               EXIT WITH 0 IF 0 INPUT
11240 -0634 00 4 11303         SXD     T1,4              SAVE IR4
11241  0774 00 4 11603         AXT     $PJ38,4           LOGXOR ATOM
11242  0634 00 4 11303         SXA     T1,4              SET FUNCTION ON PDL
11243  0074 00 4 01723         TSX     $SAVE,4
11244 -3 11305 0 01770         TXL     $END1,,T1+2       SAVE 1 ITEM
11245  0074 00 4 14225         TSX     $EVLIS,4          EVALUATE LIST OF ARGUMENTS
11246  0074 00 4 01736         TSX     UNSAVE,4
11247  0560 00 0 15244         LDQ     =0                RINGSUM OF NO ARGUMENTS
11250 -0600 00 0 11304         STQ     T1+1
11251  0560 00 0 11276         LDQ     ERS               TRA TO INSTRUCTIONS FOR INNER LOOP
                        *      COMMON PART OF LOGAND, LOGOR, AND LOGXOR
11252 -0600 00 0 11264    LOG2 STQ     LOG5
11253  0634 00 2 11272         SXA     LOG4,2            SAVE IR2
```

```
11254 -0734 00 2 00000          PDX     ,2                  POINTER TO ARGUMENT LIST
                        *       FORM THE PROPER LOGICAL COMBINATION OF THE ARGUMENTS
11255  0500 00 2 00000  LOG1    CLA     0,2                 L
11256 -0734 00 2 00000          PDX     0,2                 CDR(L)
11257  0734 00 4 00000          PAX     0,4
11260 -0754 00 4 00000          PXD     0,4                 CAR(L)
11261  0074 00 4 12567          TSX     NUMVAL,4            GET NUMBER FOR THIS ELEMENT
11262 -0734 00 4 00000          PDX     0,4
11263 -0500 00 4 00000          CAL     0,4
11264  0 00000 0 00000  LOG5            **                  INSTRUCTION SET EARLIER
11265  3 00000 2 11255          TXH     LOG1,2,0            LOOP AGAIN IF CDR(L) NOT NULL
                        *       RETURN A POINTER TO THE RESULT
11266 -0500 00 0 11304  LOG6    CAL     T1+1                PICK UP RESULT
11267  0560 00 0 00150          LDQ     $OCTD               MAKE AN OBJECT OF IT
11270  0074 00 4 11163          TSX     $MKNO,4
11271 -0534 00 4 11303          LXD     T1,4                RESTORE IR4 AND IR2
11272  0774 00 2 00000  LOG4    AXT     **,2
11273  0020 00 4 00001          TRA     1,4
                        *       INSTRUCTIONS TO BE INSERTED IN INNER LOOP
11274 -0602 00 0 11304  ORS     ORS     T1+1
11275  0320 00 0 11304  ANS     ANS     T1+1
11276  0020 00 0 11277  ERS     TRA     *+1                 TRA SINCE ERS TAKES 2 INSTRUCTIONS
11277  0322 00 0 11304          ERA     T1+1
11300  0602 00 0 11304          SLW     T1+1
11301  3 00000 2 11255          TXH     LOG1,2,0
11302  0020 00 0 11266          TRA     LOG6
                        *
11303 -000000000000      T1     OCT     -0,-0               STORAGE FOR LOGAND, ETC.
                        *       THIS ROUTINE USES NUMVAL,$MKNO,$ZERO,AND SEVENS
```

```
                        *       LEFTSHIFT(X,N)
                        *
                        *       IF N IS +, X IS SHIFTED LEFT N PLACES.
                        *       IF N IS -, X IS SHIFTED RIGHT -N PLACES.
                        *       BOTH INPUTS MUST BE NUMERICAL OBJECTS.
                        *
11305  0634 00 4 11330  LSHIFT  SXA     LSH1,4              SAVE IR4
11306  0634 00 2 11327          SXA     LSH4,2              SAVE IR2
11307  0601 00 0 11303          STO     T2                  SAVE X
11310  0131 00 0 00000          XCA
11311 -0734 00 2 00000          PDX     0,2                 FIND VALUE OF N
11312  0074 00 4 11423          TSX     FIXVAL,4
11313  0774 00 4 77100          AXT     7*4096+7*512+1*64,4  SET UP ARS
11314 -0120 00 0 11316          TMI     LSH2                IF NEGATIVE, SET UP ARS
11315  0774 00 4 76700          AXT     7*4096+6*512+7*64,4  SET UP ALS
11316 -0634 00 4 11324  LSH2    SXD     LSH3,4              PUT OP CODE INTO INSTRUCTION
11317  0621 00 0 11324          STA     LSH3                PUT ADDRESS IN SHIFT INSTRUCTION
11320  0500 00 0 11303          CLA     T2                  FIND VALUE OF X
11321  0074 00 4 12567          TSX     NUMVAL,4
11322 -0734 00 4 00000          PDX     0,4
11323 -0500 00 4 00000          CAL     0,4
```

MORE LISP FUNCTIONS

```
11324  0767 00 0 00000   LSH3   ALS    **          THIS INSTRUCTION WAS SET UP EARLIER
11325  0560 00 0 00150          LDQ    $OCTD       FORM OCTAL NUMBER
11326  0074 00 4 11163          TSX    $MKNO,4
11327  0774 00 2 00000   LSH4   AXT    **,2        RESTORE IR2
11330  0774 00 4 00000   LSH1   AXT    **,4
11331  0020 00 4 00001          TRA    1,4
                11303   T2     SYN    T1
```

MORE LISP FUNCTIONS

```
                        INSERT  $ARAYIF
                *
                *       THIS ROUTINE USES $MKNO,$OCTD,AND NUMVAL
                Q       HED
                *
                *
                * ARYGET       THE FUNCTION THAT GETS AND SETS THE VALUES OF ARRAYS
                *              USED IN LISP AS FOLLOWS ...
                *              TO GET A VALUE  (NAME,D1,D2,D3)
                *              TO SET A VALUE   (NAME,SET,VALUE,D1,D2,D3)
                *
                *              THE CALLING SEQUENCE IS  AS FOLLOWS
                *       SXA    ARYGTX,4
                *       TSX    ARYGET,4
                *       PZE    LOCATION OF TABLE 1,,NUMBER OF DIMENSIONS
                *
11332  0634 00 2 11371  ARYGET SXA    ARYY,2          SAVE INDEX REGISTERS
11333  0634 00 1 11372         SXA    ARYZ,1
11334  0601 00 0 11420         STO    AGAO            SAVE ARGUMENT 1
11335  0500 00 4 00003         CLA    3,4             TABLE ZERO PARAMETER WORD
11336  0621 00 0 11367         STA    AGXEX           ADDRESS OF END OF TABLE 1
11337 -0734 00 2 00000         PDX    0,2             NUMBER OF DIMENSIONS
11340 -0600 00 0 11421         STQ    AGAT            ARG 2
11341  0500 00 0 02531         CLA    $ARG3
11342  0601 00 0 11422         STO    AGATH           ARGUMENT 3
11343  0500 00 0 11416         CLA    AX              XEC INSTRUCTION
11344 -0534 00 4 11420         LXD    AGAO,4          GET ARG 1
11345 -3 10614 4 11357         TXL    AGN,4,$SET-1    TEST FOR SET OPERATION
11346  3 10615 4 11357         TXH    AGN,4,$SET      GO ON IF NOT $SET
11347 -0600 00 0 11417         STQ    AGV             IS SET SAVE VALUE
11350  0500 00 0 02531         CLA    $ARG3
11351  0601 00 0 11420         STO    AGAO            DIMENSION 1
11352  0500 00 0 02532         CLA    $ARG4
11353  0601 00 0 11421         STO    AGAT            DIMENSION 2
11354  0500 00 0 02533         CLA    $ARG5
11355  0601 00 0 11422         STO    AGATH           DIMENSION 3
11356  0500 00 0 11415         CLA    AXS             XEC* INSTRUCTION
11357  0622 00 0 11367   AGN   STD    AGXEX           SET UP FETCH OR STORE INSTUCTION
11360  3 00002 2 11374         TXH    AGDTH,2,2       GO IF 3 D ARRAY
11361  3 00001 2 11404         TXH    AGDT,2,1        GO IF 2 D ARRAY
11362 -0534 00 2 11420         LXD    AGAO,2          DIMENSION 1
11363  0074 00 4 11423         TSX    FIXVAL,4        EVALUATE THE FIXED POINT NUMBER
11364  0734 00 1 00000         PAX    0,1             INTO PROPER INDEX
11365  0774 00 6 00000         AXT    0,6             ZERO INDEX  REGISTERS
11366  0500 00 0 11417   AGXE  CLA    AGV             GET THE VALUE
11367  0522 00 4 00000   AGXEX XEC    **,4            FETCH BY XEC OR STORE BY XEC*
11370  0774 00 4 00000  ARYGTX AXT    **,4            RESTORE INDEX REGISTERS
11371  0774 00 2 00000   ARYY  AXT    **,2
11372  0774 00 1 00000   ARYZ  AXT    **,1
11373  0020 00 4 00001         TRA    1,4
                *
11374 -0534 00 2 11422   AGDTH LXD    AGATH,2         DIMENSION 3
11375  0074 00 4 11423         TSX    FIXVAL,4        EVALUATE AS A FIXED POINT NUMBER
11376  0734 00 1 00000         PAX    0,1             INTO INDEX
11377 -0534 00 2 11420         LXD    AGAO,2          DIMENSION 1
```

MORE LISP FUNCTIONS

```
11400  0074 00 4 11423          TSX     FIXVAL,4        EVALUATE IT
11401  0621 00 0 11413          STA     AGR             SET UP AXT INSTRUCTION
11402 -0534 00 2 11421          LXD     AGAT,2
11403  0020 00 0 11411          TRA     AGD1
                        *
11404  0634 00 0 11413  AGDT    SXA     AGR,0           PRESET AXT INSTRUCTION
11405 -0534 00 2 11421          LXD     AGAT,2
11406  0074 00 4 11423          TSX     FIXVAL,4        FIXED POINT NUMBER EVALUATION
11407  0734 00 1 00000          PAX     0,1             INTO INDEX 1
11410 -0534 00 2 11420          LXD     AGAD,2
11411  0074 00 4 11423  AGD1    TSX     FIXVAL,4
11412  0734 00 2 00000          PAX     0,2             INTO INDEX 2
11413  0774 00 4 00000  AGR     AXT     **,4            ZERO OR DIMENSION 1
11414  0020 00 0 11366          TRA     AGXE            GO BACK TO MAIN PROGRAM
                        *
11415  0522 60 0 00000  AXS     XEC*    **          THE STORE INSTRUCTION
11416  0522 00 0 00000  AX      XEC     **          THE FETCH INSTRUCTION
11417  0 00000 0 00000   AGV                          VALUE TO BE STORED PUT HERE
11420  0 00000 0 00000   AGAD                         DIMENSION 1
11421  0 00000 0 00000   AGAT                         DIMENSION 2
11422  0 00000 0 00000   AGATH                        DIMENSION 3
```

MORE LISP FUNCTIONS

```
                                    INSERT  $FIXVLF
                                *
                                *       FIXVAL
                                *
                                *       FIXVAL HAS AS INPUT A POINTER TO A FIXED POINT NUMBER OBJECT IN
                                *       IR2, AND HANDS BACK THE NUMERICAL VALUE OF THAT OBJECT.
                                *
11423   0634 00 2 11434  FIXVAL SXA     FXVE,2          SAVE IR2 IN CASE OF ERROR
11424   0500 00 2 00000          CLA     0,2
11425   0734 00 2 00000          PAX     0,2
11426  -3 77776 2 11434          TXL     FXVE,2,-2       ERROR IF NOT ATOMIC
11427  -0734 00 2 00000          PDX     0,2
11430  -0320 00 0 00122          ANA     $QT1
11431   0100 00 0 11434          TZE     FXVE
11432   0500 00 2 00000          CLA     0,2             PICK UP VALUE
11433   0020 00 4 00001          TRA     1,4             NORMAL EXIT
11434   0774 00 2 00000  FXVE    AXT     **,2            IR2 SHOULD LAND IN DECR. OF AC
11435  -0754 00 2 00000          PXD     0,2             IT DOES INDEED LAND THERE
11436                             ERROR   FIXVAL
```

```
                        INSERT  $ARAY2F
                    *
                    *
                    * ARYMAK       THE FUNCTION THAT MAKES ARRAYS
                    *              THE ARGUMENT IS A SINGLE LIST WHOSE SUB-LISTS HAVE THE
                    *              FORM  (NAME,(DIMENSION1,DIMENSION2,DIMENSION3),TYPE)
                    *       ARRAYS MAY BE 1, 2, OR 3 DIMENSIONAL AND MAY BE OF LIST OR
                    *              NON-LIST TYPE.
                    *
                    *              ARRAY IS STORED AS FOLLOWS ...
                    *       SXA    ARTGTX,4              ADDRESS OF SUBR TXL INSTRUCTION
                    *       TSX    ARYGET,4
                    *       PZE    END + 1,, N OF DIMENSIONS  (ARRAY PROPERTY POINTS HERE)
                    *       PZE    TOTAL LENGTH,,LIST LENGTH
                    *       PZE    TABLE ZERO,, NUMBER OF DIMENSIONS (ARYGET PARAMETER WORD)
                    *       CLA*   **,2                  TABLE 1
                    *       **************************
                    *       STO    **,1                  TABLE 2
                    *       ****************************
                    *                                    ARRAY PROPER GOES HERE
                    *
 11441  0560 00 0 11443  ARYMAK LDQ    AMFAG             PICK UP FUNCTIONAL ARGUMENT
 11442  0020 00 0 03201         TRA    MAPLIS            LET MAPLIST HANDLE ITERATION ALONG LIS
                    *
 11443 -3 00001 0 11444   AMFAG TXL    *+1,,1            FUNCTIONAL ARGUMENT
 11444  0634 00 4 11650         SXA    AFRX,4            SAVE INDEX REGISTERS
 11445  0634 00 2 11651         SXA    AFRY,2
 11446 -0734 00 4 00000         PDX    0,4               POINTER TO LIST
 11447  0500 00 4 00000         CLA    0,4
 11450  0734 00 4 00000         PAX    0,4               POINTER TO SUBLIST
 11451 -0500 00 4 00000         CAL    0,4
 11452  0734 00 4 00000         PAX    0,4               NAME
 11453 -0634 00 4 02664         SXD    AFAT,4            SAVE IT
 11454 -0734 00 4 00000         PDX    0,4
 11455  0500 00 4 00000         CLA    0,4
 11456  0734 00 2 00000         PAX    0,2               POINTER TO DIMENSION LIST
 11457 -0734 00 4 00000         PDX    0,4
 11460  0500 00 4 00000         CLA    0,4
 11461  0734 00 4 00000         PAX    0,4               TYPE
 11462  0600 00 0 11666         STZ    ATYP
 11463 -3 11646 4 11466         TXL    ADA,4,$LIST-1     GO IF NOT $ LIST
 11464  3 11647 4 11466         TXH    ADA,4,$LIST
 11465 -0634 00 4 11666         SXD    ATYP,4            MAKES ATYP NON-ZERO FOR LIST ARRAYS
 11466  0500 00 2 00000  ADA    CLA    0,2               FIRST WORD ON DIMENSION LIST
 11467  0734 00 2 00000         PAX    0,2               DIMENSION 1
 11470  0622 00 0 02665         STD    ATMP              POINTER TO REST
 11471  0074 00 4 11423         TSX    FIXVAL,4          EVALUATE THE FIXED POINT NUMBER
 11472  0601 00 0 11670         STO    ADO               DIMENSION 1
 11473 -0534 00 4 02665         LXD    ATMP,4            PICK UP POINTER TO REST OF LIST
 11474 -3 00000 4 11512         TXL    AOD,4,0           GO IF 1 D
 11475  0500 00 4 00000         CLA    0,4               NEXT WORD
 11476  0622 00 0 02665         STD    ATMP              SAVE POINTER
 11477  0734 00 2 00000         PAX    0,2               DIMENSION 2
 11500  0074 00 4 11423         TSX    FIXVAL,4          GET NUMBER VALUE
 11501  0601 00 0 11671         STO    ADT               DIMENSION 2
```

```
11502 -0534 00 4 02665          LXD     ATMP,4          POINTER TO REST OF LIST
11503 -3 00000 4 11521          TXL     ATD,4,0         GO IF 2 D ARRAY
11504  0500 00 4 00000          CLA     0,4
11505  0734 00 2 00000          PAX     0,2             DIMENSION 3
11506  0074 00 4 11423          TSX     FIXVAL,4        NUMBER VALUE
11507  0601 00 0 11672          STO     ADTH            DIMENSION 3
11510  0774 00 2 00003          AXT     3,2             NUMBER OF DIMENSIONS
11511  0020 00 0 11527          TRA     AGA             GO TO NEXT PART OF PROGRAM
11512  0500 00 0 11670    ADD   CLA     ADO             ID, TREAT AS A 1 X 1 X DI ARRAY
11513  0601 00 0 11672          STO     ADTH
11514  0500 00 0 15245          CLA     =1
11515  0601 00 0 11671          STO     ADT             DIMENSION 2
11516  0601 00 0 11670          STO     ADO             DIMENSION 1
11517  0774 00 2 00001          AXT     1,2             1 D ARRAY
11520  0020 00 0 11527          TRA     AGA             GO NEXT PART
11521  0601 00 0 11672    ATD   STO     ADTH
11522  0500 00 0 11670          CLA     ADO
11523  0601 00 0 11671          STO     ADT
11524  0500 00 0 15245          CLA     =1
11525  0601 00 0 11670          STO     ADO             DIMENSION 1
11526  0774 00 2 00002          AXT     2,2             2 D ARRAY
11527  0560 00 0 11670    AGA   LDQ     ADO             DIMENSION 1
11530 -0754 00 0 00000          PXD     0,0             ZERO AC
11531  0200 00 0 11671          MPY     ADT             DIMENSION 2
11532 -0600 00 0 11665          STQ     ADOT            D1 X D2
11533  0200 00 0 11672          MPY     ADTH            DIMENSION 3
11534  0520 00 0 11666          ZET     ATYP            SKIP NEXT IF NON-LIST ARRAY
11535 -0600 00 0 11666          STQ     ATYP            LIST LENGHT
11536  0131 00 0 00000          XCA                     D1 X D2 X D3 TO AC
11537  0400 00 0 11665          ADD     ADOT            ADD INDEX TABLE LENGTHS
11540  0400 00 0 11670          ADD     ADO
11541  0400 00 0 15250          ADD     =5              CONSTANT LENGTH
11542  0621 00 0 11663          STA     APWT            PARAMETER WORD TWO
11543  0621 00 0 11667          STA     ATMQ            SAVE LENGTH
11544  0534 00 4 11666          LXA     ATYP,4          ZERO OR LIST LENGTH
11545 -0634 00 4 11663          SXD     APWT,4
11546  0074 00 4 02766          TSX     BLOCKR,4        RESERVE A BLOCK OF THIS LENGTH
11547  0100 00 0 11653          TZE     ARYTL           GO IF ARRAY WILL NOT FIT
11550  0621 00 0 02665          STA     ATMP            END OF BLOCK ADDRESS
11551  0400 00 0 15245          ADD     =1              ADD 1
11552  0621 00 0 11662          STA     APWO            PARAMETER WORD 1
11553 -0634 00 2 11664          SXD     ATBZ,2          NUMBER OF DIMENSIONS
11554 -0634 00 2 11662          SXD     APWO,2
11555 -0634 00 2 11673          SXD     ASBR,2
11556  0402 00 0 11667          SUB     ATMQ            LENGTH OF BLOCK
11557  0621 00 0 11673          STA     ASBR            ADDRESS OF BEGINNING OG BLOCK
11560  0737 00 4 00000          PAC     0,4             POINTER IN IR 4
11561  1 77776 4 11562          TXI     *+1,4,-2        POINTER TO ARRAY PROPERTY
11562 -0634 00 4 11674          SXD     AARY,4          SAVE POINTER
11563  0737 00 4 00000          PAC     0,4             POINTER TO BEGINNING OF ARRAY
11564  0400 00 0 15247          ADD     =4              LENGTH OF PREFIX - 1
11565  0400 00 0 11670          ADD     ADO
11566  0621 00 0 11664          STA     ATBZ            LAST LOC. IN TABLE ONE
11567  0774 00 2 00005          AXT     5,2             LENGTH OF PREFIX TO ARRAY
11570  0500 00 2 11665    ACLA  CLA     ADOT,2          PICK UP PREFIX
```

```
11571  0601 00 4 00000          STO    0,4                 AND STORE IN CORE
11572  1 77777 4 11573          TXI    *+1,4,-1            UPDTAEC CORE LOCATION
11573  2 00001 2 11570          TIX    ACLA,2,1      GET REST OF PREFIX
11574 -0320 00 0 00132          ANA    $AMASK              TABLE ZERO IN AC
11575 -0501 00 0 11675          ORA    ACLAS               OR IN CLA* INSTRUCTION
11576  0534 00 2 11670          LXA    ADO,2               LENGTH OF TABLE
11577  0400 00 0 11671   AADD   ADD    ADT                 INCREMENT BY DIMENSION 2
11600  0601 00 4 00000          STO    0,4                 PUT IN CORE
11601  1 77777 4 11602          TXI    *+1,4,-1            UP DATE CORE COUNTER
11602  2 00001 2 11577          TIX    AADD,2,1            FINISH OFFF
11603  0534 00 2 11665          LXA    ADOT,2              LENGTH OF TABLE 2
11604 -0320 00 0 00132          ANA    $AMASK              CLEAR OUT ALL BUT ADDRESS
11605 -0501 00 0 11676          ORA    ARSTO               PUT INSTRUCTION
11606  0400 00 0 11672   AAA    ADD    ADTH                ADD DIMENSION 3
11607  0601 00 4 00000          STO    0,4                 PUT IN CORE
11610  1 77777 4 11611          TXI    *+1,4,-1            UPDATE CORE COUNTER
11611  2 00001 2 11606          TIX    AAA,2,1             CONTINUE TO CONSTRUCT TABLE
                        *               TABLE CONSTRUCTION ALL DONE.
                        *               THE FOLLOWING ADDS PROPERTYS TO THE ARYATOM
11612  0500 00 0 11674          CLA    AARY                PICK UP POINTER TC TO ARRAY PROPERTY
11613  0560 00 0 15244          LDQ    =0
11614  0074 00 4 02714          TSX    $CONS,4
11615  0560 00 0 15244          LDQ    =0
11616  0074 00 4 02714          TSX    $CONS,4
11617  0131 00 0 00000          XCA
11620  0500 00 0 11677          CLA    ARY                 POINTER TC ATCMIC SYMBOL ARRAY
11621  0074 00 4 02714          TSX    $CONS,4             (ARRAY,(POINTER TC ARRAY PROPERTY))
11622  0601 00 0 02665          STO    ATMP                SAVE IN TEMP STORAGE
11623  0500 00 0 11673          CLA    ASBR                TXL INSTRUCTIONM
11624  0074 00 4 02674          TSX    $CONSW,4            PUT IN FULL WORD SPACE
11625  0560 00 0 02665          LDQ    ATMP                REST OF PROPERTIES
11626  0074 00 4 02714          TSX    $CONS,4
11627  0131 00 0 00000          XCA
11630  0500 00 0 00153          CLA    ASB                 POINTER TC $SUBR ATCMIC SYMBOL
11631  0074 00 4 02714          TSX    $CONS,4
11632  0131 00 0 00000          XCA                        SAVE IN MQ
11633 -0534 00 4 02664          LXD    AFAT,4              POINTER TO NAME
11634  0500 00 4 00000          CLA    0,4                 FIRST WCRD
11635 -0734 00 4 00000          PDX    0,4                 SAVE POINTER TO REST
11636 -0754 00 4 00000          PXD    0,4                 PUT IN AC
11637  0131 00 0 00000          XCA                        INTER CHANGE AC AND MQ
11640  0074 00 4 07562          TSX    $NCONC,4            SPLICE  2 LISTS TOGETHER
11641 -0534 00 4 02664          LXD    AFAT,4              PUINTER TO FIRST WORD ON PROPERTY LIST
11642  0622 00 4 00000          STD    0,4                 REPLACE DECREMENT OPERATION
11643 -0754 00 4 00000          PXD    0,4                 POINTER TO ARRY ATOM
11644  0560 00 0 02515          LDQ    ARYLIS              PICK UP ARRAY LIST
11645  0074 00 4 02714          TSX    $CONS,4             PUT ON AS ACTIVE ARRAY
11646  0622 00 0 02515          STD    ARYLIS              UPDATE ARRAY LIST
11647  0500 00 0 02664          CLA    AFAT                FINAL ANSWER
11650  0774 00 4 00000   AFRX   AXT    **,4                RESTORE INDEX REGISTERS
11651  0774 00 2 00000   AFRY   AXT    **,2
11652  0020 00 4 00001          TRA    1,4                 EXIT
                        *
11653  0534 00 2 11651   ARYTL  LXA    AFRY,2              RESTORE INDEX 2
11654  0500 00 0 02664          CLA    AFAT                ARRAY NAME
```

```
11655                                  ERROR   NARYRM          NOT ENOUGH RCOM FOR ARRAY

                                   *                           CONSTANTS AND STORAGE
11660   0634 00 4 11370                SXA     ARYGTX,4        5 WORD PREFIX TO ARRAYS
11661   0074 00 4 11332                TSX     ARYGET,4
11662   0 00000 0 00000        APWO                            END+1,,N OF D
11663   0 00000 0 00000        APWT                            LENGTH,,LIST LENGTH
11664   0 00000 0 00000        ATBZ                            TABLE ZERO,, N OF D
11665   0 00000 0 00000        ADOT                            D1 X D2
11666   0 00000 0 00000        ATYP                            ZERO OR LIST LENGTH
11667   0 00000 0 00000        ATMQ                            TEMPORARY STORAGE
11670   0 00000 0 00000        ADO                             D1
11671   0 00000 0 00000        ADT                             D2
11672   0 00000 0 00000        ADTH                            D3
11673  -3 00000 0 00000        ASBR    TXL     **,,**
11674   0 00000 0 00000        AARY                            POINTER TO ARRAY PROPERTY
11675   0500 60 2 00000        ACLAS   CLA*    **,2            FETCH INSTRUCTION
11676   0601 00 1 00000        ARSTO   STO     **,1            PUT INSTRUCTION
11677   0 13223 0 00000        ARY             ,,$ARRAY
                   00153        ASB     SYN     $SUBRD
```

```
                        INSERT  $ARTHIF
                        *
                        *
                        * UNUMIX       EVALUATES ITS 2 NUMERICAL ARGUMENTS AND FLOATS THE FIXED
                        *               POINT ARGUMENT IF A MIXED EXPRESSION.  THE NUMERICAL
                        *               VALUES ARE LEFT IN AC AND MQ WITH TYPE OF NUMBER IN $ARG3
                        *
11700  0634 00 4 11721  UNUMIX SXA     UNUX,4          SAVE LINK IR
11701 -0600 00 0 11746          STQ     UNUT            SAVE SECOND ARGUMENT
11702  0074 00 4 12567          TSX     NUMVAL,4        NUMERICALLY EVALUATE THE FIRST ARG
11703 -0734 00 4 00000          PDX     0,4             POINTER TO FULL WORD
11704  0500 00 4 00000          CLA     0,4             NUMERICAL VALUE
11705  0601 00 0 11745          STO     UNUS            SAVE IT
11706 -0600 00 0 11747          STQ     UNUR            SAVE TYPE OF NUMBER
11707  0500 00 0 11746          CLA     UNUT            PICK UP SECOND ARG
11710  0074 00 4 12567          TSX     NUMVAL,4        NUMERICALLY EVALUATE IT
11711 -0734 00 4 00000          PDX     0,4             POINTER TO FULL WORD
11712  0500 00 4 00000          CLA     0,4             NUMERICAL VALUE
11713  0131 00 0 00000          XCA                     VLUE TO MQ, TYPE TO AC
11714  0402 00 0 11747          SUB     UNUR            COMPARE WITH TYPE OF FIRST
11715 -0100 00 0 11723          TNZ     UNMXA           TRA IF NOT SAME
11716  0500 00 0 11747  UNUE    CLA     UNUR            PICK UP NUMBER TYPE
11717  0601 00 0 02531          STO     $ARG3
11720  0500 00 0 11745          CLA     UNUS            PICK UP FIRST NUMERICAL VALUE
11721  0774 00 4 00000  UNUX    AXT     **,4            RESTORE LINK IR
11722  0020 00 4 00001          TRA     1,4             EXIT
                        *
11723 -0600 00 0 11746  UNMXA   STQ     UNUT            MIXED TYPES, SAVE SECOND VALUE
11724  0634 00 2 11736          SXA     UNUX2,2         SAVE IR 2
11725 -0534 00 2 11747          LXD     UNUR,2          PICK UP TYPE OF FIRST NUMBER
11726  0074 00 4 12775          TSX     FIXFLO,4        DISPATCH
11727  0761 00 0 00000          NOP                     IMPOSSIBLE RETURN
11730  0020 00 0 11740          TRA     UNMXB           FLOAT SECOND NUMBER
11731  0500 00 0 11745          CLA     UNUS            FIRST NUMBER
11732  0074 00 4 13012          TSX     $UNFIX,4        FLOAT IT
11733  0560 00 0 00143          LDQ     UNFLT           $FLOAT FOR TYPE
11734 -0600 00 0 02531          STQ     $ARG3
11735  0560 00 0 11746          LDQ     UNUT            SECOND NUMBER
11736  0774 00 2 00000  UNUX2   AXT     **,2            RESTORE IR 2
11737  0020 00 0 11721          TRA     UNUX            RESTORE LINK AND EXIT
                        *
11740  0131 00 0 00000  UNMXB   XCA                     FLOAT SECOND NUMBER
11741  0074 00 4 13012          TSX     $UNFIX,4        FLOAT FUNCTION
11742  0131 00 0 00000          XCA                     BACK TO MQ
11743  0534 00 2 11736          LXA     UNUX2,2         RESTORE IR 2
11744  0020 00 0 11716          TRA     UNUE            GET FIRST NUMBER, RESTORE LINK + EXIT
11745  0 00000 0 00000  UNUS                            FIRST NUMERICAL VALUE
11746  0 00000 0 00000  UNUT                            SECOND ARG AND VALUE
11747  0 00000 0 00000  UNUR                            TYPE OF FIRST ARG
                 00143  UNFLT   SYN FLOATD              FLOAT INDICATOR
                        *
                        * THIS ROUTINE USES NUMVAL,$UNFIX,FIXFLO, AND $ARG3  + $FLOAT
                        *
                        *
                        * DIVIDE        DIVIDES THE FIRST NUMERICAL ARGUMENT BY THE SECOND. THE
```

```
                                *               ANSWER IS A LIST OF THE QUOTIENT AND THE REMAINDER.
                                *
                                * QUOTEN        GIVES THE QUOTIENT WHEN THE FIRST NUMERICAL ARGUMENT IS
                                *               DIVIDED BY THE SECOND.
                                *
                                * REMAIN        GIVES THE REMAINDER WHEN THE FIRST NUMERICAL ARGUMENT IS
                                *               DIVIDED BY THE SECOND.
11750  0604 00 0 12040  DIVIDE STI     DIVND          SAVE INDICATORS
11751  0057 00  000003         RIR     3               DIIDE INDICATE
11752  0020 00 0 11762         TRA     DIVOP          DO OPERATION
                                *
11753  0604 00 0 12040  REMAIN STI     DIVND          SAVE INDICATORS
11754  0057 00  000003         RIR     3              DIVIDE INDICATE
11755  0055 00  000002         SIR     2              SET REMAINDER INDICATOR
11756  0020 00 0 11762         TRA     DIVOP          DO OPERATION
                                *
11757  0604 00 0 12040  QUOTEN STI     DIVND          SAVE INDICATORS
11760  0057 00  000003         RIR     3              DIVIDE INDICATE
11761  0055 00  000001         SIR     1              QUOTIENT INDICATOR
11762  0634 00 4 12031   DIVOP SXA     DIVX,4         SAVE LINK IR
11763  0634 00 2 12030         SXA     DIVX2,2        SAVE IR 2
11764  0074 00 4 11700         TSX     UNUMIX,4       NUMERICALLY EVALUATE THE ARGUMENTS
11765 -0534 00 2 02531         LXD     $ARG3,2        PICK UP TYPE
11766 -0600 00 0 12041         STQ     DIVT           SECOND ARG
11767  0074 00 4 12775         TSX     FIXFLO,4       DISPATCH ON TYPE
11770  0761 00 0 00000         NOP                    IMPOSSIBLE RETURN
11771  0241 00 0 12041         FDP     DIVT           FLOATING DIVIDE
11772  0020 00 0 12021         TRA     DIVFX          DO FIXED POINT DIVIDE
11773  0760 00 0 00012  DIVDC DCT                     CHECK  FOR ILLEGAL DIVISION
11774  0074 00 4 01624         TSX     $DCT,4         DIVIDE CHECK ERROR
11775  0054 00  000001         RFT     1              SEE IF REMAINDER IS TO BE SAVED
11776  0020 00 0 12026         TRA     DIVA           NO, SET UP QUOTIENT
11777 -0600 00 0 12041         STQ     DIVT           YES, SAVE QUOTEINT
12000  0560 00 0 02531         LDQ     $ARG3          PICK UP TYPE
12001  0074 00 4 11163         TSX     $MKNO,4        MAKE REMAINDER A NUMBER
12002  0054 00  000002         RFT     2              TEST TO SEE IF QUOTIENT IS WANTED
12003  0020 00 0 12034         TRA     DIVEX          NO, RESTORE AND EXIT
12004  0560 00 0 15244         LDQ     =0             NIL IN MQ
12005  0074 00 4 02714         TSX     $CONS,4        LIST OF REMAINDER
12006  0131 00 0 00000         XCA                    SHUTTLE INTO MQ
12007  0500 00 0 12041         CLA     DIVT           PICK UP QUOTIENT
12010 -0600 00 0 12041         STQ     DIVT           SAVE LIST OF REMAINDER
12011  0560 00 0 02531         LDQ     $ARG3          PICK UP TYPE
12012  0074 00 4 11163         TSX     $MKNO,4        MAKE QUOTIENT A NUMBER
12013  0560 00 0 12041         LDQ     DIVT           LIST(REMAINDER)
12014  0074 00 4 02714         TSX     $CONS,4        LIST(QUOTIENT,REMAINDER)
12015  0534 00 4 12031         LXA     DIVX,4         RESTORE LINK IR
12016  0534 00 2 12030         LXA     DIVX2,2        RESTORE IR 2
12017  0441 00 0 12040         LDI     DIVND          RESTORE INDICATORS
12020  0020 00 4 00001         TRA     1,4            EXIT
                                *
12021  0131 00 0 00000  DIVFX XCA                     FIXED POINT DIVISION, PUT ARG 1 IN MQ
12022 -0754 00 0 00000         PXD     0,0            CLEAR AC
12023  0763 00 0 00000         LLS     0              MQ SIGN TO AC
12024  0221 00 0 12041         DVP     DIVT           DIVIDE BY ARG 2
```

```
12025  0020 00 0 11773          TRA     DIVDC           PREFORM DIVIDE CHECK AND CARRY ON
12026  0131 00 0 00000  DIVA    XCA                     QUOTIENT TO AC
12027  0560 00 0 02531          LDQ     $ARG3           TYPE TO MQ
12030  0774 00 2 00000  DIVX2   AXT     **,2            RESTORE IR 2
12031  0774 00 4 00000  DIVX    AXT     **,4            RESTORE LINK IR
12032  0441 00 0 12040          LDI     DIVND           RESTORE INDICATORS
12033  0020 00 0 11163          TRA     $MKNO
                        *
12034  0534 00 2 12030  DIVEX   LXA     DIVX2,2         EXIT ROUTINE, RESTORE IR 2
12035  0534 00 4 12031          LXA     DIVX,4          RESTORE LINK IR
12036  0441 00 0 12040          LDI     DIVND           RESTORE INDICATORS
12037  0020 00 4 00001          TRA     1,4
                        *
12040  0 00000 0 00000  DIVND                           INDICATORS STORAGE
12041  0 00000 0 00000  DIVT                            LIST AND NON-LIST TEMPORARY STORAGE
                        *
                        * THIS ROUTINE USES $MKNO,$DCT,$CONS,$ARG3 AND UNUMIX
                        *
                        *
                        * DIFFER        COMPUTES THE DIFFERENCE BETWEEN ITS 2 NUMERICAL ARGUMENTS
                        *
12042  0634 00 4 12055  DIFFER  SXA     DIFX,4          SAVE LINK IR
12043  0634 00 2 12054          SXA     DIFX2,2         SAVE IR 2
12044  0074 00 4 11700          TSX     UNUMIX,4        NUMERICALLY EVALUATE THE ARGUMENTS
12045 -0534 00 2 02531          LXD     $ARG3,2         PICK UP TYPE OF NUMBERS
12046 -0600 00 0 12057          STQ     DIFT            STORE SECOND NUMBER
12047  0074 00 4 12775          TSX     FIXFLO,4        DISPATCH ON TYPE
12050  0761 00 0 00000          NOP                     IMPOSSIBLE RETURN
12051  0302 00 0 12057          FSB     DIFT            FLOATING POINT
12052  0402 00 0 12057          SUB     DIFT            FIXED POINT
12053  0560 00 0 02531          LDQ     $ARG3           TYPE OF NUMBER
12054  0774 00 2 00000  DIFX2   AXT     **,2            RESTORE IR 2
12055  0774 00 4 00000  DIFX    AXT     **,4            RESTORE LINK IR
12056  0020 00 0 11163          TRA     $MKNO           MAKE RESULT A NUMBER
                        *
12057  0 00000 0 00000  DIFT                            TEMPORARY STORAGE
```

MORE LISP FUNCTIONS

```
                        INSERT  $EXPTFE
                *       THIS VERSION OF EXPT IS BASED UPON THE SHARE
                *   EXP(1, EXP(2, AND EXP(3 CURRENT AT HARVARD AS OF DECEMBER 19, 1965.
                *   THE PREVIOUS VERSION WAS A HOPELESS PIECE OF 704 CODE.

                        HEAD    Q               HEY YOU GUYS.. HEADING IS Q

12060  0634 00 4 12116  EXPT    SXA     EXPTX,4         SAVE MACHINE CONDITIONS
12061  0601 00 0 00044          STO     PTRBS           POINTER TO BASE ATOM
12062  0131 00 0 00000          XCA                     POINTER TO EXPONENT IN AC
12063  0074 00 4 12567          TSX     NUMVAL,4        WANT EXP IN AC, TYPE IN MQ
12064 -0734 00 4 00000          PDX     0,4             POINTER TO EXP IN X4
12065  0500 00 4 00000          CLA     0,4             EXP TO AC
12066  0601 00 0 00045          STO     EXPNT           ..
12067  0131 00 0 00000          XCA                     TYPE.(EXP) IN AC
12070  0600 00 0 00046          STZ     FLTEXP          ASSUME IS FIXED
12071  0340 00 0 00143          CAS     $FLOATD         BUT IS IT REALLY
12072  0020 00 0 12074          TRA     *+2             *YES
12073 -0625 00 0 00046          STL     FLTEXP          *NO

12074  0500 00 0 00044          CLA     PTRBS           NOW SORT OUT THE BASE
12075  0074 00 4 12567          TSX     NUMVAL,4        WANT BASE IN AC, TYPE IN MQ
12076 -0734 00 4 00000          PDX     0,4             POINTER TO BASE IN X4
12077  0500 00 4 00000          CLA     0,4             BASE TO AC
12100  0131 00 0 00000          XCA                     VICE VERSA
12101  0340 00 0 00143          CAS     $FLOATD         IS BASE FLOATING OR FIXED
12102  0020 00 0 12104          TRA     *+2             *FIXED
12103  0020 00 0 12120          TRA     BASFLT          *FLOATING

                *       HERE WHEN BASE IS FIXED

12104  0600 00 0 00047          STZ     FLTBAS          SET SWITCH FOR FIXED BASE
12105  0131 00 0 00000          XCA                     BRING BASE BACK TO AC
12106  0560 00 0 00045          LDQ     EXPNT           BRING EXPONENT TO MQ
12107  0162 00 0 12112          TQP     POSEXP          IF EXPONENT IS POSITIVE, OK
12110  0074 00 4 13012          TSX     $UNFIX,4        BUT OTHERWISE, MUST HAVE FLOATING BASE
12111  0020 00 0 12121          TRA     BSFLTA          ..
12112  0520 00 0 00046  POSEXP  ZET     FLTEXP          WHAT SORT OF THING IS EXPONENT
12113  0020 00 0 12130          TRA     FXFLT           *FLOATING
12114  0074 00 4 12277          TSX     EXP(12,4        *  FIXED**FIXED
12115  0560 00 0 00142          LDQ     $FIXD           ANSWER IS FIXED TOO
12116  0774 00 4 00000  EXPTX   AXT     **,4            RETURN TO USER VIA SIMULATED TSX  MKNO,4
12117  0020 00 0 11103          TRA     $MKNO

                *       HERE IF BASE IS FLOATING

12120  0131 00 0 00000  BASFLT XCA                     BASE TO AC
12121 -0625 00 0 00047  BSFLTA STL     FLTBAS          SET SWITCH FOR FLOATING BASE
12122  0520 00 0 00046          ZET     FLTEXP          WHAT SORT OF THING IS EXPONENT
12123  0020 00 0 12131          TRA     FLTFLT          *  FLOATING**FLOATING
12124  0560 00 0 00045          LDQ     EXPNT           * FLOATING**FIXED
12125  0074 00 4 12277          TSX     EXP(12,4        ..
12126  0560 00 0 00143  EXPFL   LDQ     $FLOATD         RESULT IS FLOATING
12127  0020 00 0 12116          TRA     EXPTX           RETURN VIA MKNO
```

```
12130  0074 00 4 13012  FXFLT  TSX     $UNFIX,4      IF FIX**FLOAT, FLOAT THE BASE
12131  0100 00 0 12126  FLTFLT TZE     EXPFL         MAKING IT FLOAT**FLOAT
12132  0760 00 0 00003         SSP                   BASE ZERO TAKEN CARE OF, NOW ALSO .L. 0
12133  0074 00 4 12221         TSX     LOG,4         A**B IS E**(B*LOG A)
12134  0131 00 0 00000         XCA                   ..
12135  0260 00 0 00045         FMP     EXPNT         ..
12136  0074 00 4 12140         TSX     EXP,4
12137  0020 00 0 12126         TRA     EXPFL         RETURN VIA MKNO


                00044  PTRBS  SYN     COMMON        POINTER TO BASE ATOM
                00045  EXPNT  SYN     COMMON+1      EXPONENT (NUMBER, NOT ATOM)
                00046  FLTEXP SYN     COMMON+2      NONZERO IFF EXPONENT IS FLOATING
                00047  FLTBAS SYN     COMMON+3      NONZERO IFF BASE IS FLOATING
                00050  M      SYN     COMMON+4      TEMPORARY FOR E-TO-THE-X FUNCTION
                00051  F      SYN     COMMON+5      ..
                00052  T      SYN     COMMON+6      ..
                00053  SQ     SYN     COMMON+7      ..
                00054  CMMON  SYN     COMMON+8      TEMPORARIES (3) FOR NATURAL LOG ROUTINE
                00057  POWER  SYN     COMMON+11     TEMPORARY FOR FIXED-EXPONENT ROUTINE
                00060  FACTOR SYN     COMMON+12     ..
```

MORE LISP FUNCTIONS

```
                        *                E TO THE X
12140  0601 00 0 00050   EXP    STO M                  STORE ARGUMENT
12141  0560 00 0 12211          LDQ MAXX                TEST IF OUT OF RANGE
12142  0040 00 0 12206          TLQ      LARGE         IF TOO LARGE ERROR RETURN
12143  0760 00 0 00002          CHS
12144  0040 00 0 12204          TLQ T1                 IF TOO SMALL RETURN WITH ZERO
12145  0560 00 0 12213          LDQ      CHAR
12146  0765 00 0 00033          LRS 27
12147 -0760 00 0 00003          SSM
12150  0400 00 0 12213          ADD CHAR               CONVERT TO FIXED POINT
12151  0621 00 0 12153          STA SH1
12152  0200 00 0 12212          MPY LOGE               X TIMES LOG E BASE 2
12153  0765 00 0 00000   SH1    LRS **                 SEPARATE INTEGER AND FRACTION
12154  0767 00 0 00033          ALS 27
12155  0601 00 0 00050          STO M
12156  0765 00 0 00004          LRS 4
12157 -0600 00 0 00051          STQ F
12160  0200 00 0 00051          MPY F                  COMPUTE FRACTION SQUARED
12161  0601 00 0 00053          STO SQ                 COMPUTE CONTINUED FRACTION
12162  0400 00 0 12215          ADD A                  8,27 SCALING
12163  0601 00 0 00052          STO T                  8,27
12164  0500 00 0 12216          CLA B                  12,23
12165  0221 00 0 00052          DVP T                  8,27 4,31
12166 -0600 00 0 00052          STQ T                  4,31
12167  0560 00 0 00053          LDQ SQ                 8,27
12170  0200 00 0 12217          MPY C                  0,35 8,27
12171  0763 00 0 00004          LLS 4                  4,31
12172  0402 00 0 00052          SUB T                  4,31
12173  0400 00 0 12220          ADD D                  4,31
12174  0402 00 0 00051          SUB F                  4,31
12175  0601 00 0 00052          STO T                  4,31
12176  0500 00 0 00051          CLA F                  5,31 NUMERATOR EQUALS 2F
12177  0560 00 0 15275          LDQ      =1.0
12200  0225 33 0 00052          VDP T,0,27
12201  0131 00 0 00000          XCA
12202 -0501 00 0 12214          ORA CH1
12203  0300 00 0 15275          FAD      =1.0
12204  0400 00 0 00050   T1     ADD M
12205  0020 00 4 00001          TRA 1,4                RETURN
12206  0500 00 0 12210   LARGE  CLA      HUGE
12207  0020 00 4 00001          TRA      1,4

12210 +377777777777      HUGE   OCT      377777777777
12211 +207540071260      MAXX   DEC 88.028
12212 -270524354513       LOGE  DEC -1.44269504098l
12213 +000000000242      CHAR   OCT 242
12214 +201000000000       CH1   OCT 201000000000
12215 +127325604305        A    DEC 87.417497202B8
12216 +046477071523        B    DEC 617.9722695B12
12217 +010676467774        C    DEC .03465735903B0
12220 +237214030720        D    DEC 9.9545957821B4
```

49230

```
                                *             NATURAL LOG ROUTINE
12221  0100 00 4 00001      LOG TZE 1,4
12222  0760 00 0 00003          SSP
12223  0560 00 0 12275          LDQ      CU            CLEAR MQ                        *
12224  0765 00 0 00033          LRS 27                 CHARACTERISTIC...I+128
12225  0767 00 0 00001          ALS 1
12226  0402 00 0 12275          SUB CU
12227 -0501 00 0 12265          ORA K2                 FLOAT I
12230  0601 00 0 00054          STO CMMON
12231 -0754 00 0 00000          ZAC
12232  0131 00 0 00000          XCA
12233  0771 00 0 00001          ARS 1                  B1
12234  0400 00 0 12266          ADD K3                 F+HLFSQ2...B1
12235  0601 00 0 00055          STO CMMON+1
12236  0402 00 0 12267          SUB K4                 F-HLFSQ2...B1
12237  0225 35 0 00055          VDP CMMON+1,0,29       1-1+35-29...B6,   Z
12240 -0600 00 0 00055          STQ CMMON+1            B6
12241  0204 34 0 00055          VLM CMMON+1,0,28       6+6+28-35...B5,    Z SQUARE
12242  0601 00 0 00056          STO CMMON+2
12243  0560 00 0 12274          LDQ VLMP               B28
12244  0204 06 0 00056          VLM CMMON+2,0,6        28+5+6-35...B4
12245  0402 00 0 12271          SUB      AA
12246  0602 00 0 00056          SLW CMMON+2
12247  0502 00 0 12272          CLS      BB
12250  0225 40 0 00056          VDP CMMON+2,0,32       5-4+35-32...B4
12251  0131 00 0 00000          XCA                    B4
12252  0402 00 0 00056          SUB CMMON+2
12253  0400 00 0 12273          ADD      DD
12254  0131 00 0 00000          XCA
12255  0765 00 0 00007          LRS 7                  BQ11
12256  0204 32 0 00055          VLM CMMON+1,0,26       11+6+26-35...B8
12257 -0501 00 0 12270          ORA K5
12260  0300 00 0 12270          FAD K5
12261  0300 00 0 00054          FAD CMMON
12262  0131 00 0 00000          XCA
12263  0260 00 0 12276          FMP LOG2
12264  0020 00 4 00001          TRA 1,4

                                  . . . . . . . . . . CONSTANTS  . . . . . . . .
12265 +232000000000          K2 OCT 232000000000
12266 +132404746316          K3 DEC .70710678118781
12267 +265011714634          K4 DEC .70710678118780
12270 +200000000000          K5 OCT 200000000000
12271 +004116753015     AA      DEC      .25963855429B4
12272 -002073464641     BB      DEC      -.26455155161B5
12273 +042011037513     DD      DEC      2.126106178B4
12274 +000000000031       VLMP OCT 000000000031
12275 +000000000401     CU      OCT 000000000401
12276 +200542710277       LOG2 DEC 6.93147180560E-01
                                . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

MORE LISP FUNCTIONS

```
                        *              ANYTHING**FIXED EXPONENTIATION ROUTINE

12277  0100 00 4 00001  EXP12 TZE  1,4                    EXIT FOR ZERO BASE
12300  0634 00 2 12351        SXA    EXP2X2,2
12301  0774 00 2 00000        AXT    0,2          ASSUME POSITIVE EXPONENT (IN MQ)
12302  0162 00 0 12304        TQP    *+2          ..
12303  0774 00 2 77777        AXT    -1,2         SET NEGATIVE MQ INDICATOR
12304  0601 00 0 00057        STO  POWER                  INITIALIZATION
12305  0500 00 0 15275        CLA    =1.0
12306 -0520 00 0 00047        NZT    FLTBAS       WE HAVE ASSUMED THAT BASE IS FLOATING
12307  0500 00 0 15245        CLA    =1           *BUT IT ISN'T
12310 -0520 00 0 00045        NZT  EXPNT
12311  0020 00 0 12351        TRA    EXP2X2       THAT 1 IS OUR ANSWER IF EXPNT IS 0
12312  0601 00 0 00060        STO  FACTOR                 X
12313  0500 00 0 00045  CLA1  CLA  EXPNT                  IS POWER
12314  0760 00 0 00001        LBT                         A FACTOR
12315  0020 00 0 12326        TRA    CLA2         *NO
12316  0560 00 0 00057        LDQ  POWER                  YES
12317 -0520 00 0 00047        NZT    FLTBAS       MAY WE USE FMP
12320  0020 00 0 12324        TRA    FXMPY        *NO
12321  0260 00 0 00060        FMP  FACTOR
12322  0601 00 0 00060        STO  FACTOR
12323  0020 00 0 12326        TRA    CLA2
12324  0200 00 0 00060  FXMPY MPY    FACTOR       SAME SEQUENCE IN FIXED POINT
12325 -0600 00 0 00060        STQ    FACTOR       ..
12326  0500 00 0 00045  CLA2  CLA  EXPNT
12327  0771 00 0 00001        ARS  1
12330  0601 00 0 00045        STO  EXPNT
12331  0100 00 0 12343        TZE  TXL-1
12332  0560 00 0 00057        LDQ  POWER
12333 -0520 00 0 00047        NZT    FLTBAS       MAY WE USE FMP
12334  0020 00 0 12340        TRA    FXMP2        *NO
12335  0260 00 0 00057        FMP  POWER
12336  0601 00 0 00057        STO  POWER
12337  0020 00 0 12313        TRA  CLA1
12340  0200 00 0 00057  FXMP2 MPY    POWER        SAME SEQUENCE IN FIXED POINT
12341 -0600 00 0 00057        STQ    POWER        ..
12342  0020 00 0 12313        TRA    CLA1         ..
12343  0500 00 0 00060        CLA  FACTOR
12344 -3 00000 2 12351   TXL  TXL  EXP2X2,2,0     *EXPONENT WAS POSITIVE
12345  0500 00 0 15275        CLA    =1.0
12346  0241 00 0 00060        FDP  FACTOR
12347 -0754 00 0 00000        ZAC
12350  0763 00 0 00043        LLS  35
12351  0774 00 2 00000  EXP2X2 AXT    **,2
12352  0020 00 4 00001        TRA  1,4
```

```
                                        INSERT  $ARTHMF
                                        HEAD    Q
                                * ADD   ADDS A STRING OF FIXED POINT OR FLOATING POINT NUMBERS
12353 -0634 00 4 02660    ADDP  SXD     AMIR,4          SAVE LINK IR
12354  0774 00 4 11251          AXT     $PLUS,4
12355  0604 00 0 02661          STI     AMIND           SAVE INDICATORS
12356  0057 00  000177         RIR     177             RESET FIRST 7 INDICATORS
12357  0055 00  000001         SIR     1               SET ADD INDICATOR (1)
12360  0020 00 0 12402          TRA     AMMMF           GO TO MAIN FUNCTION
                                *
12361 -0634 00 4 02660    MULT  SXD     AMIR,4          SAVE LINK IR
12362  0774 00 4 10416          AXT     $TIMES,4
12363  0604 00 0 02661          STI     AMIND           SAVE INDICATORS
12364  0057 00  000177         RIR     177             RESET FIRST 7 INDICATORS
12365  0055 00  000002         SIR     2               SET MULTIPLY INDICATOR (2)
12366  0020 00 0 12402          TRA     AMMMF           GO TO MAIN FUNCTION
                                *
12367 -0634 00 4 02660    MIN   SXD     AMIR,4          SAVE LINK IR
12370  0774 00 4 11513          AXT     $MINP,4
12371  0604 00 0 02661          STI     AMIND           SAVE INDICATORS
12372  0057 00  000177         RIR     177             RESET FIRST 7 INDICATORS
12373  0055 00  000010         SIR     10              SET MINIMUM FUNCTION INDICATOR (10)
12374  0020 00 0 12402          TRA     AMMMF           GO TO MAIN FUNCTION
                                *
12375 -0634 00 4 02660    MAX   SXD     AMIR,4          SAVE LINK IR
12376  0774 00 4 11524          AXT     $MAXP,4
12377  0604 00 0 02661          STI     AMIND           SAVE INDICATORS
12400  0057 00  000177         RIR     177             RSET FIRST 7 INDICATORS
12401  0055 00  000004         SIR     4               SET MAXIMUN FUNCTION INDICATOR (4)
12402  0634 00 4 02660  AMMMF  SXA     AMIR,4          PUT PROGRAM NAME WITH LINK IR
12403  0074 00 4 01723          TSX     $SAVE,4         OTHER 3 FUNCTIONS ENTER AT *-1
12404 -3 02663 0 01766          TXL     $END2,,AMIND+2  SAVE 2 ITEMS
12405  0074 00 4 14225          TSX     $EVLIS,4        EVALUATE THE LIST OF ARGUMENTS
12406  0074 00 4 01736          TSX     UNSAVE,4        RESTORE IR 4 AND INDICATORS
12407  0634 00 2 12563          SXA     AMIR2,2         SAVE IR 2
12410  0600 00 0 12566          STZ     AMSUM           ZERO FINAL ANSWER REGISTER
12411 -0734 00 4 00000  AMLP  PDX     0,4             PUT POINTER TO ARG LIST IN IR 4
12412 -3 00000 4 12555          TXL     AMEND,4,0       GO TO EXIT IF NULL
12413  0500 00 4 00000          CLA     0,4             GET FIRST WORD
12414  0601 00 0 02662          STO     AMLIS           SAVE THE WORD
12415  0734 00 4 00000          PAX     0,4             CAR OF LIST
12416 -0754 00 4 00000          PXD     0,4             TO DECREMENT
12417  0074 00 4 12567          TSX     NUMVAL,4        EVALUATE THE ITEM
12420 -0600 00 0 02663          STQ     AMQ             SAVE CHARACTERISTIC ($FIX OR $FLOAT)
12421  0056 00  000100         RNT     100             TEST FOR FIRST TIME THROUGH
12422  0020 00 0 12445          TRA     AMFRS           IS FIRST TIME GO TO INITIALIZE AMSUM
12423  0054 00  000002         RFT     2               TEST FOR MULT FUNCTION
12424  0020 00 0 12465          TRA     AMLT            EXECUTE MULT FUNCTION
12425 -0734 00 4 00000          PDX     0,4             POINTER TO FULL WORD
12426  0500 00 4 00000          CLA     0,4             GET NUMERICAL VALUE
12427  0056 00  000001         RNT     1               SKIP NEXT INSTRUCTION IF ADD FUNCTION
12430  0020 00 0 12525          TRA     AMM             EXECCUTE MAX OR MIN FUNCTION
12431 -0534 00 2 02663          LXD     AMQ,2           ADD FUNCTION. PICK UP TYPE OF NUMBER
12432  0074 00 4 12775          TSX     FIXFLO,4        TEST FOR FIX OR FLOAT
12433  0761 00 0 00000          NOP                     IMPOSSIBLE RETURN
```

```
12434  0020 00 0 12460          TRA    AFLL     EXECUTE FAD
12435  0055 00  000020          SIR    20       IS FIXED POINT. SET FIXED POINT IND.
12436 -0774 00 4 12462          AXC    AFLR,4   PRESET IR 4
12437  0054 00  000040          RFT    40       SKIP NEXT INSTRUCTION IF NOT MIXED EXP
12440  0020 00 0 12513          TRA    UNFX     IS MIXED, FLOAT THIS NUMBER
12441  0400 00 0 12566          ADD    AMSUM    FIXED ADD OF SUM
12442  0601 00 0 12566  AMRT    STO    AMSUM    STORE NEW SUM
12443  0500 00 0 02662          CLA    AMLIS    PICK UP ARG LIST
12444  0020 00 0 12411          TRA    AMLP     DO NEXT ITEM
12445 -0734 00 4 00000  AMFRS   PDX    0,4      POINTER TO FULL WORD
12446  0500 00 4 00000          CLA    0,4      GET NUMERICAL VALUE
12447  0601 00 0 12566          STO    AMSUM    STORE NUMERICAL VALUE IN AMSUM
12450 -0534 00 2 02663          LXD    AMQ,2    PICK UP TYPE OF NUMBER
12451  0074 00 4 12775          TSX    FIXFLO,4 TEST FOR FIX OR FLOAT
12452  0761 00 0 00000          NOP             IMPOSSIBLE EXIT
12453  0055 00  000040          SIR    40       SET FLOAT INDICATOR
12454  0055 00  000020          SIR    20       SET FIX INDICATOR
12455  0055 00  000100          SIR    100      SET INDICATOR SO IT WILL NOT GET BACK
12456  0500 00 0 02662          CLA    AMLIS    PICK UP REST OF ARG LIST
12457  0020 00 0 12411          TRA    AMLP     DO NEXT ITEM
12460  0055 00  000040  AFLL    SIR    40       IS FLOATING POINT, SET PROPER INDICATO
12461  0054 00  000020          RFT    20       SKIP NEXT INSTRUCTION IF NOT MIXED EXP
12462  0074 00 4 12515  AFLR    TSX    MIXFL,4  UNMIX THE EXPRESSION
12463  0300 00 0 12566          FAD    AMSUM    FLOATING ADD THE CURRENT SUM
12464  0020 00 0 12442          TRA    AMRT     STORE AND DO NEXT ITEM ON LIST
12465 -0734 00 4 00000  AMLT    PDX    0,4      POINTER TO FULL WORD
12466  0500 00 4 00000          CLA    0,4      GET NUMERICAL VALUE
12467 -0534 00 2 02663          LXD    AMQ,2    PICK UP TYPE
12470  0074 00 4 12775          TSX    FIXFLO,4 TEST FOR FIX OR FLOAT
12471  0761 00 0 00000          NOP             IMPOSSIBLE RETURN
12472  0020 00 0 12505          TRA    AFMP     DO FMP
12473  0055 00  000020          SIR    20       SET FIXED POINT INDICATOR
12474 -0774 00 4 12507          AXC    AFLT,4   PRESET IR 4
12475  0054 00  000040          RFT    40       SKIP NEXT INSTRUCTION IF NOT MIXED EXP
12476  0020 00 0 12513          TRA    UNFX     IS MIXED, FLOAT THIS NUMBER
12477  0131 00 0 00000          XCA             NUMBER TO MQ
12500  0200 00 0 12566          MPY    AMSUM    MPY BY CURRENT ANSWER
12501  0131 00 0 00000          XCA             PUT LEAST SIGNIFICANT BITS IN AC
12502  0601 00 0 12566  AMRU    STO    AMSUM    STORE NEW ANSWER
12503  0500 00 0 02662          CLA    AMLIS    PICK UP ARG LIST
12504  0020 00 0 12411          TRA    AMLP     DO NEXT ITEM
12505  0055 00  000040  AFMP    SIR    40       SET FLOATING POINT INDICATOR
12506  0054 00  000020          RFT    20       TEST FOR MIXED EXP
12507  0074 00 4 12515  AFLT    TSX    MIXFL,4  UNMIX THE EXPRESSION
12510  0131 00 0 00000          XCA             NUMBER TO MQ
12511  0260 00 0 12566          FMP    AMSUM    FMP BY CURRENT ANSWER
12512  0020 00 0 12502          TRA    AMRU     STORE NEW ANSER AND DO NEXT ITEM
12513  0057 00  000020  UNFX    RIR    20       RESET FIXED POINT INDICATOR
12514  0020 00 0 13012          TRA    $UNFIX   FLOAT THE NUMBER IN THE AC
12515  0634 00 4 12523  MIXFL   SXA    MXIR,4   FIX MIXED EXPRESSION
12516  0601 00 0 12565          STO    AMR      SAVE AC
12517  0500 00 0 12566          CLA    AMSUM    PICK UP CURRENT ANSWER
12520  0074 00 4 12513          TSX    UNFX,4   FLOAT IT
12521  0601 00 0 12566          STO    AMSUM    PUT IT AWAY
12522  0500 00 0 12565          CLA    AMR      RESTORE AC
```

```
12523  0774 00 4 00000   MXIR  AXT   **,4          RESTORE IR 4
12524  0020 00 4 00001         TRA   1,4           RETURN
12525 -0534 00 2 02663   AMM   LXD   AMQ,2         MAX OR MIN FUNCTION. GET TYPE
12526  0074 00 4 12775         TSX   FIXFLO,4      TEST FOR FIX OR FLOAT
12527  0761 00 0 00000         NOP                 IMPOSSIBLE RETURN
12530  0020 00 0 12543         TRA   AFL           EXECUTE FLOATING SECTION
12531  0055 00  000020         SIR   20            SET FIXED POINT INDICATOR
12532  0054 00  000040         RFT   40            TEST FOR MIXED EXP
12533  0074 00 4 12513         TSX   UNFX,4        FLOAT THE ARGUMENT IF MIXED
12534  0056 00  000004  AMRNT  RNT   4             TEST FORMAX FUNCTION
12535  0020 00 0 12547         TRA   AMIN          EXECUTE MIN FUNCTION
12536  0340 00 0 12566         CAS   AMSUM         COMPARE WITH CURRENT ANSWER
12537  0601 00 0 12566         STO   AMSUM         IS GREATER, STORE AS NEW ANSWER
12540  0761 00 0 00000         NOP                 THEY ARE EQUAL
12541  0500 00 0 02662         CLA   AMLIS         IS LESS, PICK UP ARGUMENT LIST
12542  0020 00 0 12411         TRA   AMLP          DO NEXT ITEM
12543  0055 00  000040  AFL    SIR   40            SET FLOATING POINT INDICATOR
12544  0054 00  000020         RFT   20            TEST FOR MIXED EXPRESSION
12545  0074 00 4 12515         TSX   MIXFL,4       UNMIX THE EXPRESSION
12546  0020 00 0 12534         TRA   AMRNT         COMPARE AND DO NEXT ITEM
12547  0340 00 0 12566  AMIN   CAS   AMSUM         MIN FUNCTION, COMPARE WITH CURRENT VAL
12550  0020 00 0 12553         TRA   *+3           IS GREATER
12551  0020 00 0 12553         TRA   *+2           IS EQUAL
12552  0601 00 0 12566         STO   AMSUM         IS LESS, STORE AS NEW ANSWER
12553  0500 00 0 02662         CLA   AMLIS         PICK UP NEXT ITEM
12554  0020 00 0 12411         TRA   AMLP          EXECUTE IT
12555  0500 00 0 12566  AMEND  CLA   AMSUM         ALL DONE. PICK UP CURRENT ANSWER
12556  0560 00 0 00142         LDQ   AMFXC         PRESET MQ
12557  0054 00  000040         RFT   40            SKIP NEXT IF FIXED POINT
12560  0560 00 0 00143         LDQ   AMFLC         PICK UP FIX IN MQ
12561  0441 00 0 02661         LDI   AMIND         RESTORE INDICATORS
12562 -0534 00 4 02660         LXD   AMIR,4        RESTORE IR 4
12563  0774 00 2 00000  AMIR2  AXT   **,2          RESTORE IR 2
12564  0020 00 0 11163         TRA   $MKNO         MAKE THE ANSWER A NUMBER
                00143  AMFLC  SYN   FLOATD        FLAOT CONSTANT
                00142  AMFXC  SYN   $FIXD         FIX CONSTANT
12565  0 00000 0 00000  AMR                       TEMP STORAGE
12566  0 00000 0 00000  AMSUM                     CURRENT ANSWER STORAGE
                        * NUMVAL       NUMERICAL VALUE TAKES ANY LIST AND DECIDES IF IT
                        *      REPRESENTS A FIXED POINT OR FLOATING POINT NUMBER. IF IT DOES NOT
                        *      THE ROUTINE CLEARS THE AC AND MQ DOES AN XEC 1,4 AND THEN GOES
                        *      TO ERROR WITH A BAD ARGUMENT COMPLAINT. IF THE LIST DOES
                        *      REPRESENT A NUMBER, UPON EXIT THE FOLLOWING THINGS ARE LEFT
                        *      AS INDICATED                POINTER TO FULL WORD IN AC
                        *                                  $FIX OR $FLOAT IN MQ
12567  0634 00 4 12617  NUMVAL SXA   NVIR4,4       SAVE LINK IR
12570  0601 00 0 02531         STO   $ARG3         SAVE ORIGINAL ARGUMENT
12571 -0734 00 4 00000         PDX   0,4           POINTER TO NUMBER IN IR 4
12572 -3 00000 4 12576         TXL   NVNO,4,0      NULL  LIST IS NOT A NUMBER
12573  0500 00 4 00000         CLA   0,4           FIRST ELEMENT
12574  0734 00 4 00000         PAX   0,4           CAR LIST
12575  3 77776 4 12607         TXH   NVATM,4,-2    GO IF AN ATOM
                        *
12576 -0754 00 0 00000  NVNO   PXD   0,0           IS NOT NUMBER, CLEAR AC
12577  0131 00 0 00000         XCA                 PUT IN MQ
```

MORE LISP FUNCTIONS

```
12600 -0754 00 0 00000          PXD     0,0             CLEAR AC AGAIN
12601  0534 00 4 12617          LXA     NVIR4,4         RESTORE LINK IR
12602  0522 00 4 00001          XEC     1,4             EXECUTE POSSIBLE EXIT INSTRUCTION
12603  0500 00 0 02531          CLA     $ARG3           MUST BE AN ERROR, PICK UP ORIGINAL ARG
12604                           ERROR   NUMVAL
                        *
12607 -0734 00 4 00000  NVATM   PDX     0,4
12610 -0320 00 0 00137          ANA     TAGMSK
12611  0100 00 0 12576          TZE     NVNO
12612  0771 00 0 00017          ARS     15
12613  0621 00 0 12615          STA     *+2
12614 -0754 00 4 00000          PXD     0,4
12615 -0774 00 4 00000          AXC     **,4
12616  0560 00 4 12620          LDQ     NVTBL,4
12617  0774 00 4 00000  NVIR4   AXT     **,4            RESTORE IR 4
12620  0020 00 4 00001  NVTBL   TRA     1,4
12621  0 12135 0 00000                  0,,$FIX
12622  0 12116 0 00000                  0,,$FLOAT
12623  0 00000 0 00000                  0,,0
12624  0 00000 0 00000                  0,,0
12625  0 12135 0 00000                  0,,$FIX
                        *
                        *
                        * ADD1          ADD 1 ADDS ONE TO ANY FIXED POINT OR FLOATING POINT
                        *       NUMBER AND EXITS WITH THE NEW NUMBER
12626  0634 00 1 12644   ADD1   SXA     A1IR1,1         SAVE IR 1
12627  0774 00 1 00000          AXT     0,1             ZERO IR 1(INDICATES ADD OP)
12630  0634 00 2 12645   ADI    SXA     A1IR2,2         SAVE IR 2
12631  0634 00 4 12646          SXA     A1IR4,4         SAVE LINK IR
12632  0074 00 4 12567          TSX     NUMVAL,4        EVALUTE NUMERICAL ARGUMENT
12633 -0600 00 0 12650          STQ     A1T             SAVE $FIX OR $FLOAT
12634 -0734 00 4 00000          PDX     0,4             POINTER TO FULL WORD
12635  0500 00 4 00000          CLA     0,4             GET NUMERICAL VALUE
12636 -0534 00 2 12650          LXD     A1T,2           PICK UP $FIX OR $FLOAT
12637  0074 00 4 12775          TSX     FIXFLO,4
12640  0761 00 0 00000          NOP                     IMPOSSIBLE RETURN
12641  0522 00 1 12651          XEC     FAD,1           IS FLOAT, DO FLOATING POINT OP
12642  0522 00 1 12653          XEC     ADDF,1          DO FIXED POINT OP
12643  0560 00 0 12650          LDQ     A1T             RESTORE $FLOAT AFTER FAD
12644  0774 00 1 00000  A1IR1   AXT     **,1            RESTORE IR 1
12645  0774 00 2 00000  A1IR2   AXT     **,2            RESTORE IR 2
12646  0774 00 4 00000  A1IR4   AXT     **,4            RESTORE LINK IR
12647  0020 00 0 11163          TRA     $MKNO           MAKE RESULT A NUMBER
                        *
12650  0 00000 0 00000  A1T                             TEMPORARY STORAGE
12651  0300 00 0 15275  FAD     FAD     =1.0            FLOATING ADD FOR ADD1
12652  0302 00 0 15275          FSB     =1.0            FOR SUB1
12653  0400 00 0 15245  ADDF    ADD     =1              FOR ADD1
12654  0402 00 0 15245          SUB     =1              FOR SUB1
                        *
                        * SUB1          SUBTRACT 1 SUBTRACTS ONE FROM A FIXED POINT OR FLOATING
                        *       POINT NUMBER. USES CODING OF ADD1 WITH AN INITIALIZATION.
12655  0634 00 1 12644   SUB1   SXA     A1IR1,1         SAVE IR 1
12656  0774 00 1 77777          AXT     -1,1            SET FOR SUBTRACT OPERATIONS
12657  0020 00 0 12630          TRA     ADI             PERFORM ADD1 CODING
```

```
                                * SUB1 USES THE CODING OF ADD1
                                *
                                *
12660  0634 00 4 12664  GRTRTP  SXA     GRTIR,4          SAVE LINK IR
12661  0074 00 4 11700          TSX     UNUMIX,4         EVALUATE NUMERICAL ARGUMENTS
12662  0040 00 0 12666          TLQ     GRTT             PREDICATE TRUE
12663 -0754 00 0 00000          PXD     0,0              FALSE, CLEAR AC
12664  0774 00 4 00000   GRTIR  AXT     **,4
12665  0020 00 4 00001          TRA     1,4              EXIT
                                *
12666  0500 00 0 00126   GRTT   CLA     $QD1             GET TRUE VALUE
12667  0020 00 0 12664          TRA     GRTIR            RESTORE LINK IR AND EXIT
                                *
                                *
                                * LESSTP       LESS THAN PREDICATE.  SIMPLE DOES GREATER THAN  PREDICATE
                                *       WITH THA ARGUMENT REVERSED.
                                *
12670  0131 00 0 00000  LESSTP  XCA                      INTERCHANGE ARGUMENTS
12671  0020 00 0 12660          TRA     GRTRTP           DO GREATER THAN PREDICATE
                                *
                                * THE FOLLOWING IS A NUMBER PREDICATE PACKAGE WHICH INCLUDES NUMBER
                                *       PREDICATE, ZERO PREDICATE, MINUS PREDICATE, ONE PREDICATE, FIX
                                *       PREDICATE AND FLOAT PREDICATE. ALL THESE PREDICATES SHARE CERTAIN
                                *       BLOCKS OF CODING AND TEMPORARY STORAGE.
                                * NUMBRP       NUMBER PREDICATE TESTS ITS ARGUMENT FOR A NUMBER.
12672  0634 00 4 12676  NUMBRP  SXA     NPIR,4           SAVE LINK IR
12673  0074 00 4 12567          TSX     NUMVAL,4         EVALUATE ARGUMENT
12674  0100 00 0 12676          TZE     NPIR             IF ZERO NOT A NUMBER
12675  0500 00 0 00126   NPT    CLA     $QD1             IS A NUMBER, PICK UP TRUTH
12676  0774 00 4 00000   NPIR   AXT     **,4             RESTORE LINK IR
12677  0020 00 4 00001          TRA     1,4              EXIT
                                *
                                * FLOATP      FLOATING POINT NUMBER PREDICATE TESTS TO SEE IF ITS
                                *       ARGUMENT IS A FLOATING POINT NUMBER
12700  0634 00 4 12676  FLOATP  SXA     NPIR,4           SAVE LINK IR
12701  0634 00 2 12752          SXA     ZPIR,2           SAVE IR 2
12702  0074 00 4 12567          TSX     NUMVAL,4         EVALUATE ARGUMENT
12703  0131 00 0 00000          XCA                      GET TYPE IN AC
12704 -0734 00 2 00000          PDX     0,2              TYPE IN IR 2
12705  0074 00 4 12775          TSX     FIXFLO,4         TEST FOR $FIX OR $FLOAT
12706  0761 00 0 00000          NOP                      IMPOSSIBLE RETURN
12707  0020 00 0 12711          TRA     FLT              IS FLOATING POINT
12710  0020 00 0 12751          TRA     ZPF              IS NOT FLOATING POINT, EXIT FALSE
12711  0500 00 0 00126   FLT    CLA     $QD1             GET TRUTH VALUE
12712  0020 00 0 12752          TRA     ZPIR             RESTORE IR S AND EXIT
                                *
                                * FIXP         FIXED POINT PREDICATE TESTS FOR FIXED POINT NUMBERS.
12713  0634 00 4 12676   FIXP   SXA     NPIR,4           SAVE LINK IR
12714  0634 00 2 12752          SXA     ZPIR,2           SAVE IR 2
12715  0074 00 4 12567          TSX     NUMVAL,4         EVALUATE ARGUMENT
12716  0131 00 0 00000          XCA                      GET TYPE IN AC
12717 -0734 00 2 00000          PDX     0,2              TYPE IN IR 2
12720  0074 00 4 12775          TSX     FIXFLO,4         TEST FOR $FIX OR $FLOAT
12721  0761 00 0 00000          NOP                      IMPOSSIBLE EXIT
12722  0020 00 0 12751          TRA     ZPF              IS FLOAT, EXIT FALSE
```

```
12723  0500 00 0 00126          CLA     $QD1            IS FIX, GET TRUTH VALUE
12724  0020 00 0 12752          TRA     ZPIR            RESTORE IR S AND RETURN
                        *
                        * MINUSP        MINUS PREDICATE TESTS TO SEE IF ITS ARGUMENT IS A
                        *       NEGATIVE NUMBER.
12725  0634 00 4 12676  MINUSP  SXA     NPIR,4          SAVE LINK IR
12726  0074 00 4 12567          TSX     NUMVAL,4        EVALUATE ARGUMENT
12727 -0734 00 4 00000          PDX     0,4
12730  0500 00 4 00000          CLA     0,4             PICK UP NUMBER
12731 -0120 00 0 12675          TMI     NPT             EXIT TRUE IF MINUS
12732 -0754 00 0 00000          PXD     0,0             IS NOT, EXIT FALSE
12733  0020 00 0 12676          TRA     NPIR            RESTORE LINK IR AND EXIT
                        *
                        * ZEROP         ZERO PREDICATE TESTS ITS ARGUMENT FOR A FIXED POINT
                        *       ZERO    OR
                        *       ZERO OR A FLOATING POINT ZERO + OR - A TOLERENCE (FLOTOL).
12734  0634 00 4 12676   ZEROP  SXA     NPIR,4          SAVE LINK IR
12735  0634 00 2 12752          SXA     ZPIR,2          SAVE IR 2
12736  0074 00 4 12567          TSX     NUMVAL,4        EVALUATE ARGUMENT
12737 -0734 00 4 00000          PDX     0,4             GET POINTER TO IR 4
12740  0500 00 4 00000          CLA     0,4             FULL WORD
12741  0760 00 0 00003  ZPG    SSP                     GET MAGNITUDE OF N
12742  0100 00 0 12754          TZE     ZPT             EXIT TRUE IF ZERO
12743  0131 00 0 00000          XCA                     PUT NUMBER IN MQ
12744 -0734 00 2 00000          PDX     0,2             PUT TYPE IN IR 2
12745  0500 00 0 13050          CLA     FLOTOL          PICK UP FLOATING POINT TOLERENCE
12746  0074 00 4 12775          TSX     FIXFLO,4        TEST FOR FIX OR FLOAT
12747  0020 00 0 12756          TRA     ZPTS            NOT FIX OR FLO MEANS FLO FROM ONEP
12750  0020 00 0 12756          TRA     ZPTS            IS FLOATING POINT, COMPARE WITH FLOTOL
12751 -0754 00 0 00000  ZPF     PXD     0,0             IS FIXED POINT, EXIT FALSE
12752  0774 00 2 00000  ZPIR    AXT     **,2            RESTORE IR 2
12753  0020 00 0 12676          TRA     NPIR            RESTORE IR 4 AND EXIT
12754  0500 00 0 00126  ZPT     CLA     $QD1            GET TRUTH VALUE
12755  0020 00 0 12752          TRA     ZPIR            RESTORE IR S AND EXIT
12756  0040 00 0 12754  ZPTS    TLQ     ZPT             IS FLOATING POINT, EXIT TRUE IF LESS
12757  0020 00 0 12751          TRA     ZPF             OTHERWISE EXIT FALSE
                        * ONEP          ONE PREDICATE TESTS TO SEE IF ITS ARGUMENT IS ONE
                        *       BY SUBTRACTING ONE AND TESTING THE RESULT WITH ZEROP.
12760  0634 00 4 12676   ONEP   SXA     NPIR,4          SAVE LINK IR
12761  0634 00 2 12752          SXA     ZPIR,2          SAVE IR 2
12762  0074 00 4 12567          TSX     NUMVAL,4        EVALUATE ARGUMENT
12763 -0734 00 4 00000          PDX     0,4             POINTER TO AC
12764  0500 00 4 00000          CLA     0,4             FULL WORD TO AC
12765  0131 00 0 00000          XCA                     TYPE TO AC
12766 -0734 00 2 00000          PDX     0,2             TYPE TO IR 2
12767  0131 00 0 00000          XCA
12770  0074 00 4 12775          TSX     FIXFLO,4        DISPATCH ON FIX OR FLOAT
12771  0761 00 0 00000          NOP                     IMPOSSIBLE RETURN
12772  0302 00 0 15275          FSB     =1.0
12773  0402 00 0 15245          SUB     =1              SUBTRACT 1
12774  0020 00 0 12741          TRA     ZPG             APPLY ZERO PREDICATE
                        *
                        *       FIXFLO                  SUBROUTINE TO DISPATCH ON FIX OR FLO,
                        *                               ARGUMENT IN IR 2.
                        *
```

```
12775 -3 12134 2 12777  FIXFLO TXL     *+2,2,$FIX-1          TXL - TXL FILTER FOR $FIX
12776 -3 12135 2 13011         TXL     FX,2,$FIX             GO IF $FIX
12777 -3 12115 2 13001         TXL     *+2,2,$FLOAT-1        TXL - TXL FILTER FOR $FLOAT
13000 -3 12116 2 13005         TXL     FL,2,$FLOAT           GO IF $FLOAT
13001  0522 00 4 00001         XEC     1,4                   EXCEUTE IF NEITHER FIX OR FLOAT
13002  0020 00 4 00004         TRA     4,4                   RETURN
13003  0020 00 4 00005         TRA     5,4                   SKIP EXIT
13004  0020 00 4 00006         TRA     6,4                   SKIP 2 EXIT
13005  0522 00 4 00002   FL    XEC     2,4                   EXECUTE IF $FLOAT
13006  0020 00 4 00004         TRA     4,4                   RETURN
13007  0020 00 4 00005         TRA     5,4                   SKIP EXIT
13010  0020 00 4 00006         TRA     6,4                   SKIP 2 EXIT
13011  0020 00 4 00003   FX    TRA     3,4
                       *       FIXFLO USES $FIX AND $FLAOT
                       * UNFIX         UNFIX MAKES A FIXED POINT ARGUMENT IN THE AC A FLOATING
                       *       POINT NUMBER LEFT IN AC. MQ IS PRESERVED.
13012  0601 00 0 13046  UNFIX STO     UFC                   SAVE ARGUMENT
13013 -0320 00 0 00121         ANA     UFMSK                 MASK OUT ALL BUT CHARACTERISTIC
13014 -0100 00 0 13023         TNZ     UFE                   IF ANY THING LEFT IT MUST BE NORMALIZD
13015  0500 00 0 13046         CLA     UFC                   NOTHING LEFT, RESTORE ARGUMENT
13016 -0501 00 0 00120         ORA     UFMC                  OR IN CHARACTERISTIC
13017 -0600 00 0 13045         STQ     UFQ                   SAVE MQ
13020  0300 00 0 00120         FAD     UFMC                  ESSENTIALLY FAD OF ZERO TO NORMALIZE
13021  0560 00 0 13045         LDQ     UFQ                   RESTORE MQ
13022  0020 00 4 00001         TRA     1,4                   EXIT
                       *
13023  0634 00 4 13043  UFE    SXA     UFXR,4                NUMBER GREATER THAN 2 TO 27. SAVE IR4
13024  0774 00 4 00234         AXT     2*64+3*8+4,4           CHARACTERISTIC SO FAR
13025  0600 00 0 13047         STZ     UFS                   INITIALIZE SIGN PORTION
13026  0120 00 0 13031         TPL     UFF                   SKIP IF +
13027  0760 00 0 00003         SSP                           MAKE IT +
13030 -0625 00 0 13047         STL     UFS                   RECORD FACT BY MAKING UFS NON-ZERO
13031  0771 00 0 00001  UFF    ARS     1                     DIVIDE NUMBER BY 2
13032  0340 00 0 00117         CAS     UFNC                  SEE IF NORMALIZED YET
13033  1 00001 4 13031         TXI     UFF,4,1               ADD 1 TO CHARACTERISTIC AND TRY AGAIN
13034  1 00001 4 13031         TXI     UFF,4,1               DITTO
13035  0601 00 0 13046         STO     UFC                   IS NORMALIZED
13036 -0754 00 4 00000         PXD     0,4                   CHARACTERISTIC TO AC
13037  0767 00 0 00011         ALS     9                     POSITION CHARACTERISTIC
13040 -0501 00 0 13046         ORA     UFC                   OR IN NORMALIZED NUMBER
13041  0520 00 0 13047         ZET     UFS                   TEST FOR SIGN, 0 MEANS +
13042 -0760 00 0 00003         SSM                           NOT ZERO SO MAKE MINUS
13043  0774 00 4 00000  UFXR   AXT     **,4                  RESTORE IR 4
13044  0020 00 4 00001         TRA     1,4                   EXIT
                        00121  UFMSK   SYN Q777Q9            CHARACTERISTIC MASK
                        00120  UFMC    SYN Q233Q9            GENERAL CHARACTERISTIC
                        00117  UFNC    SYN $Q01Q9
13045  0 00000 0 00000  UFQ                                  MQ
13046  0 00000 0 00000  UFC                                  AC TEMPORARY STORAGE
13047  0 00000 0 00000  UFS                                  SIGN STORAGE
                       * UNFIX USES NO EXTERNAL CONSTANTS.
                       * FLOTOL        FLOATING POINT TOLERENCE USED IN DESIDING IF FLOATING
                       *       POINT NUMBERS ARE INTEGERS.
13050 +156622516334    FLOTOL DEC      3E-6                  FLOATING POINT TOLERENCE VALUE
                       * MNSPRG        MINUS PROGRAM MAKES A LIST OF MINUS AND ITS ARGUMENT
```

```
                                *
                                * MNSPRG       CREATES A NUMBER OF OPPOSITE SIGN OF NUMERICAL ARGUMENT
                                *
13051  0634 00 4 13056  MNSPRG  SXA    MRXR,4          SAVE LINK IR
13052  0074 00 4 12567          TSX    NUMVAL,4        EVALUATE THE NUMERICAL ARGUMENT
13053 -0734 00 4 00000          PDX    0,4             POINTER TO FULL WORD
13054  0500 00 4 00000          CLA    0,4             NUMERICAL VALUE
13055  0760 00 0 00002          CHS                    MAKE OPPOSITE SIGN
13056  0774 00 4 00000  MRXR    AXT    **,4            RESTORE LINK IR
13057  0020 00 0 11163          TRA    $MKNO           MAKE IT A NUMBER
                                *
                                * RCPPRG       CALCULATES THE RECIPORICAL OF A NUMBER.
13060  0634 00 4 13101  RCPPRG  SXA    RRXR,4          SAVE LINK IR
13061  0634 00 2 13102          SXA    RRXR2,2         SAVE IR 2
13062  0074 00 4 12567          TSX    NUMVAL,4        EVALUTE THE NUMERICAL ARGUMENT
13063 -0734 00 4 00000          PDX    0,4             POINTER TO FULL WORD
13064  0500 00 4 00000          CLA    0,4             NUMERICAL VALUE
13065  0601 00 0 13107          STO    RCPT            SAVE VALUE
13066  0131 00 0 00000          XCA                    TYPE TO AC
13067 -0734 00 2 00000          PDX    0,2             TYPE TO IR 2
13070  0074 00 4 12775          TSX    FIXFLO,4        DISPATCH ON FIX OR FLOAT
13071  0761 00 0 00000          NOP                    IMPOSSIBLE RETURN
13072  0500 00 0 15275          CLA    =1.0            IS FLOAT, PICK UP FLOATING POINT 1
13073  0020 00 0 13104          TRA    RCPFX           IS FIXED POINT
13074  0241 00 0 13107          FDP    RCPT            DIVIDE BY ARGUMENT
13075  0760 00 0 00012          DCT                    CHECK FOR ILLEGAL DIVISION
13076  0074 00 4 01624          TSX    $DCT,4          DIVIDE CHECK ERROR
13077  0131 00 0 00000          XCA                    QUOTENT TO AC
13100  0560 00 0 00143          LDQ    RCPS            $FLOAT TO MQ
13101  0774 00 4 00000  RRXR    AXT    **,4            RESTORE LINK IR
13102  0774 00 2 00000  RRXR2   AXT    **,2            RESTORE IR 2
13103  0020 00 0 11163          TRA    $MKNO           MAKE ANSWER A NUMBER
                                *
13104  0131 00 0 00000  RCPFX   XCA                    FIXED POINT RECIP, ANSWER IS ZERO
13105 -0754 00 0 00000          PXD    0,0             CLEAR AC
13106  0020 00 0 13101          TRA    RRXR            RESTORE IR S AND MAKE A NUMBER
                                *
13107  0 00000 0 00000  RCPT                          TEMPORARY STORAGE
                  00143  RCPS    SYN    FLOATD          FLOAT INDICATOR
                                *
                                INSERT $APPLIF
```

```
                              APPLY

                              APPLY(F,L,A) =
                                 SELECT(CAR(L).,
                                        -1,APP2(F,L,A).,
                                        LAMBDA,EVAL(F,APPEND(PAIR(CADR(F),L),A)).,
                                        LABEL,APPLY(CADDR(F),L,APPEND(
                                             PAIR1(CADR(F),CADDR(F))),A).,
                                        APPLY(EVAL(F,A),L,A))

                          A       HED
 13110 -0634 00 4 02560    APPLY SXD ASS1,4
 13111  0100 00 4 00001          TZE 1,4
 13112  0601 00 0 02563          STO AST1                 F
 13113 -0734 00 4 00000          PDX 0,4
 13114  0634 00 4 02560          SXA     ASS1,4           SAVE FUNCTION ALONG WITH INDEX REGISTE
 13115  0500 00 4 00000          CLA 0,4                  CWR(F)
 13116  0734 00 4 00000          PAX 0,4                  CAR(F)
 13117  3 77776 4 13147          TXH ASP1,4,-2            =-1
 13120 -0754 00 4 00000          PXD 0,4
 13121  0340 00 0 00147          CAS ASLMD                = LAMBDA
 13122  0020 00 0 13124          TRA *+2
 13123  0020 00 0 13152          TRA ASP2
 13124  0340 00 0 00145          CAS ASFUN
 13125  0020 00 0 13127          TRA *+2
 13126  0020 00 0 13225          TRA ASP4
 13127  0340 00 0 00146          CAS ASLBL                = LABEL
 13130  0020 00 0 13132          TRA *+2
 13131  0020 00 0 13175          TRA ASP3
 13132  0074 00 4 01723          TSX $SAVE,4
 13133 -3 02564 0 01764          TXL     $END3,,ASSA+2    SAVE 3 ITEMS
 13134 -0600 00 0 02561          STQ ASSL
 13135  0560 00 0 02531          LDQ $ARG3
 13136 -0600 00 0 02562          STQ ASSA
 13137  0500 00 0 02563          CLA AST1                 F
 13140  0074 00 4 13703          TSX $EVAL,4              EVAL(F,A)
 13141  0560 00 0 02562          LDQ ASSA
 13142 -0600 00 0 02531          STQ $ARG3
 13143  0560 00 0 02561          LDQ ASSL
 13144  0074 00 4 01736          TSX UNSAVE,4
 13145 -0534 00 4 02560          LXD ASS1,4
 13146  0020 00 0 13110          TRA APPLY                APPLY(EVAL(F,A),L,A)

 13147  0500 00 0 02563    ASP1  CLA AST1                 F
 13150 -0534 00 4 02560          LXD ASS1,4
 13151  0020 00 0 13243          TRA $APP2                P APP2(F,L,A)
                          *          LAMBDA BRANCH
 13152 -0534 00 4 02563    ASP2  LXD AST1,4               F
 13153  0500 00 0 02531          CLA $ARG3
 13154  0601 00 0 02565          STO AST3
 13155  0500 00 4 00000          CLA 0,4                  CWR(F)
 13156 -0734 00 4 00000          PDX 0,4                  CDR(F)
 13157  0500 00 4 00000          CLA 0,4                 CWDR(F)
 13160  0601 00 0 02566          STO AST4
 13161  0734 00 4 00000          PAX 0,4                  CADR(F)
```

```
13162 -0754 00 4 00000          PXD 0,4
13163  0074 00 4 07447          TSX $PAIR,4                 PAIR(CADR(F),L)
13164  0560 00 0 02565          LDQ AST3                    A
13165  0074 00 4 07562          TSX     $NCONC,4
13166  0131 00 0 00000          XCA
13167 -0534 00 4 02566          LXD AST4,4                  CDDR(F)
13170  0500 00 4 00000          CLA 0,4
13171  0734 00 4 00000          PAX 0,4
13172 -0754 00 4 00000          PXD 0,4
13173 -0534 00 4 02560          LXD ASS1,4
13174  0020 00 0 13703          TRA $EVAL      EVAL(CADDR(F),APPEND(PAIR(CADR(F),L),A))

                        *               LABEL BRANCH
13175 -0534 00 4 02563   ASP3   LXD AST1,4                  F
13176 -0600 00 0 02564          STQ AST2                    L
13177  0560 00 0 02531          LDQ $ARG3                   A
13200 -0600 00 0 02565          STQ AST3
13201  0500 00 4 00000          CLA 0,4                     CWR(F)
13202 -0734 00 4 00000          PDX 0,4                     CDR(F)
13203  0500 00 4 00000          CLA 0,4
13204  0601 00 0 02566          STO AST4                    CWDR(F)
13205 -0734 00 4 00000          PDX 0,4                     CDDR(F)
13206  0500 00 4 00000          CLA 0,4
13207  0734 00 4 00000          PAX 0,4                     CADDR(F)
13210 -0754 00 4 00000          PXD 0,4
13211  0601 00 0 02563          STO AST1
13212  0131 00 0 00000          XCA
13213  0534 00 4 02566          LXA AST4,4
13214 -0754 00 4 00000          PXD 0,4                     CADR(F)
13215  0074 00 4 02714          TSX $CONS,4                 CONS(CADR(F),CONS(CADDR(F),0))
13216  0560 00 0 02565          LDQ AST3                    A
13217  0074 00 4 02714          TSX $CONS,4                 APPEND( ABOVE,A)
13220  0601 00 0 02531          STO $ARG3
13221  0560 00 0 02564          LDQ AST2
13222  0500 00 0 02563          CLA AST1                    CADDR(F)
13223 -0534 00 4 02560          LXD ASS1,4
13224  0020 00 0 13110          TRA APPLY    APPLY(CADDR(F),L,APPEND(PAIR(CADR(F),CADDR(F)),A))

                        *               FUNARG BRANCH
13225 -0534 00 4 02563   ASP4   LXD AST1,4                  F
13226  0500 00 4 00000          CLA ,4
13227 -0734 00 4 00000          PDX ,4                      CDR(F)
13230  0500 00 4 00000          CLA ,4
13231  0601 00 0 02563          STO AST1                    CWDR(F)
13232 -0734 00 4 00000          PDX ,4                      CDDR(F)
13233  0500 00 4 00000          CLA ,4
13234  0734 00 4 00000          PAX ,4                      CADDR(F)
13235 -0754 00 4 00000          PXD ,4
13236  0601 00 0 02531          STO $ARG3                   A
13237  0534 00 4 02563          LXA AST1,4                  CADR(F)
13240 -0754 00 4 00000          PXD ,4                      F
13241 -0534 00 4 02560          LXD ASS1,4
13242  0020 00 0 13110          TRA $APPLY
```

```
                    00146   ASLBL SYN LABELD
                    00147   ASLMD SYN LAMDAD
                    00145   ASFUN SYN     FNARGD

                              APP2(F,L,A)=SELECT(F.,CAR,CAAR(L).,CDR,
                              CDAR(L).,CONS,CONS(CAR(L),CADR(L)).,LIST,COPY(L).,SEARCH(F,
                              LAMBDA(J,CAR(J)=SUBR OR CAR(J)=EXP),
                              LAMBDA(J,CAR(J)=SUBR YIELDS APP3(CWADR
                              (J),DISTRIB(L)),1 YIELDS APPLY(CADR(J),L,A)))
                              ERROR)

                                    A       HED
13243 -0634 00 4 13401      APP2    SXD     ATS1,4          SAVE LINK IR
13244 -0734 00 4 00000              PDX     ,4              SAVE NAME OF FUNCTION IN ADDR
13245  0500 00 0 13401              CLA     ATS1            NAME FOR TRACING
13246  0634 00 4 13401              SXA     ATS1,4          SAVE IR4 AND
13247  0601 00 0 02525              STO     CSV
13250 -0754 00 4 00000              PXD     ,4              RESTORE THE FN TO AC
13251 -0534 00 4 02531              LXD     $ARG3,4         GET ALIST
13252 -0634 00 4 13405              SXD     A,4             SAVE IT
13253 -0600 00 0 13404              STQ     AL              ARGUMENT LIST
13254  0601 00 0 13403              STO     F               FUNCTION  (IS ATOMIC SYMBOL)
13255  0600 00 0 13377              STZ     APTRT           INITIALIZE TRACE TEST CELL
13256 -0734 00 4 00000      APSES   PDX     0,4             ARG TO IR
13257 -3 00000 4 13321              TXL     APSAL,4,0       GO IF NO MORE PROPERTY LIST
13260  0500 00 4 00000              CLA     0,4             FIRST WORD
13261  0734 00 4 00000              PAX     0,4             CAR
13262 -3 10404 4 13264              TXL     *+2,4,$TRACE-1
13263 -3 10405 4 13351              TXL     APTRK,4,$TRACE  LOOK FOR TRACE
13264 -3 10464 4 13266              TXL     *+2,4,$SUBR-1   LOOK FOR
13265 -3 10465 4 13300              TXL     R2,4,$SUBR      $SUBR OR
13266 -3 12246 4 13256              TXL     APSES,4,$EXPR-1 $EXPR
13267  3 12247 4 13256              TXH     APSES,4,$EXPR
                              *             EXPR BRANCH IN APPLY
13270 -0734 00 4 00000              PDX     0,4             POINTER TO NEXT WORD AFTER $EXPR
13271  0500 00 4 00000              CLA     0,4             NEXT WORD
13272  0734 00 4 00000              PAX     0,4             CAR
13273 -0754 00 4 00000              PXD     0,4             IS FUNCTION
13274  0520 00 0 13377              ZET     APTRT           TEST FOR TRACE MODE
13275  0020 00 0 13341              TRA     APTXP           TRACE THIS EXPRESSION
13276 -0534 00 4 13401              LXD     ATS1,4          RESTORE LINK IR
13277  0020 00 0 13110              TRA     $APPLY          GO TO APPLY
                              *             R2 THE SUBR BRANCH OF APPLY.
13300 -0734 00 4 00000      R2      PDX     0,4             GET THE TXL INSTRUCTION BT  TAKING
13301  0500 00 4 00000              CLA     0,4             CWR (CADR L))
13302  0734 00 4 00000              PAX     0,4
13303  0500 00 4 00000              CLA     0,4
13304  0601 00 0 13400              STO     CWADR           TXL INSTRUCTION
13305  0500 00 0 13404              CLA     AL              GET THE ARGUMENT LIST
13306  0074 00 4 07713              TSX     SPREAD,4        SPREAD IT INTO AC, MQ, ARG3, ETC.
13307  0520 00 0 13377              ZET     APTRT           TEST FOR TRACE MODE
13310  0020 00 0 13363              TRA     APTSB           TRACE THIS SUBROUTINE
13311  0074 00 4 01723              TSX     $SAVE,4
13312 -3 02530 0 01766              TXL     $END2,,$ALIST+2
13313 -0534 00 4 13405              LXD     A,4
```

```
13314 -0634 00 4 02526          SXD     $ALIST,4
13315  0074 00 4 13400          TSX     CWADR,4
13316  0074 00 4 01736          TSX     UNSAVE,4
13317 -0534 00 4 02525          LXD     CSV,4
13320  0020 00 4 00001          TRA     1,4
                        *
13321  0500 00 0 13402  APSAL   CLA     FAS             WHERE TO GO IF NOT FOUND ON PAIR LIST
13322  0601 00 0 02531          STO     $ARG3
13323  0500 00 0 13403          CLA     F               ATOMIC FUNCTION
13324  0560 00 0 13405          LDQ     A
13325  0074 00 4 07663          TSX     SASSOC,4        SEARCH PAIR LIST FOR LABEL DEFINITION
13326 -0734 00 4 00000          PDX     0,4             POINTER TO ASSOCIATED ITEM
13327  0500 00 4 00000          CLA     0,4
13330 -0734 00 4 00000          PDX     0,4             POINTER TO ITEM
13331 -0754 00 4 00000          PXD     0,4
13332  0560 00 0 13405          LDQ     A               RESTORE PAIR LIST
13333 -0600 00 0 02531          STQ     $ARG3
13334  0560 00 0 13404          LDQ     AL              RESTORE ARGUMENT LIST
13335  0520 00 0 13377          ZET     APTRT           TEST FOR TRACE MODE
13336  0020 00 0 13341          TRA     APTXP           TRACE THIS EXPRESSION
13337 -0534 00 4 13401          LXD     ATS1,4          RESTORE LINK IR
13340  0020 00 0 13110          TRA     $APPLY          GO TO APPLY WITH ITEM ASSOCIATED WITH
                        *                               THE ATOMIC FUNCTION
13341  0074 00 4 01723  APTXP   TSX     $SAVE,4         TRACE EXPR
13342 -3 02527 0 01770          TXL     $END1,,CSV+2
13343  0074 00 4 13110          TSX     $APPLY,4
13344  0020 00 0 13370          TRA     APEXC           FINISH UP
                        *
13345  0500 00 0 13403  R33     CLA     F               PICK UP FUNCTION
13346                           ERROR   UNDFFN          FUNCTION OBJECT HAS NO DEFINITION
                        *
13351 -0625 00 0 13377  APTRK   STL     APTRT
13352  0601 00 0 13376          STO     APA             SAVE THE AC
13353  0500 00 0 02560          CLA     ASS1
13354  0601 00 0 02525          STO     CSV
                        ****INSERTION ABOVE***********
13355  0534 00 4 02560          LXA     ASS1,4          ATOM NAME
13356 -0754 00 4 00000          PXD     0,4
13357  0074 00 4 14301          TSX     ARGOF,4         PRINT ARGUMENTS OF
13360  0560 00 0 13404          LDQ     AL              RESTORE MQ AFTER PRINTING
13361  0500 00 0 13376          CLA     APA             RESTORE AC
13362  0020 00 0 13256          TRA     APSES           CONTINUE PROPERTY LIST SEARCH
                        *
13363  0074 00 4 01723  APTSB   TSX     $SAVE,4         TRACE SUBR
13364 -3 02530 0 01766          TXL     $END2,,$ALIST+2
13365 -0534 00 4 13405          LXD     A,4
13366 -0634 00 4 02526          SXD     $ALIST,4
13367  0074 00 4 13400          TSX     CWADR,4
13370  0074 00 4 01736  APEXC   TSX     UNSAVE,4
13371  0131 00 0 00000          XCA                     VALUE TO MQ
13372  0534 00 4 02525          LXA     CSV,4
13373 -0754 00 4 00000          PXD     0,4             TO AC
13374 -0534 00 4 02525          LXD     CSV,4
13375  0020 00 0 14335          TRA     VALOF           PRINT VALUE OF
                        *
```

APPLY AND EVAL

```
13376  0 00000 0 00000   APA                                AC STORAGE
13377  0 00000 0 00000   APTRT                              TRACE MODE TEST SWITCH
13400  0 00000 0 00000   CWADR                              TXL INSTRUCTION FOR SUBR
13401  0 00000 0 00000   ATS1                               LINK INDEX REGISTER
13402 -3 00000 0 13345   FAS    TXL      R33,,0             NOT FOUND ON PAIR LIST SC CALL ERROR
13403  0 00000 0 00000   F                                  ATOMIC FUNCTION GOES HERE
13404  0 00000 0 00000   AL                                 ARGUMENT LIST
13405  0 00000 0 00000   A                                  A OR PAIR LIST
                         *

                         A      HED
13406  0100 00 0 13703   EVCON  TZE $EVAL
13407 -0634 00 4 02572          SXD ECS1,4
13410  0074 00 4 01723          TSX $SAVE,4
13411 -3 02577 0 01762          TXL      $END4,,ECS4+2      SAVE 4 ITEMS
13412 -0600 00 0 02573          STQ ECS2
13413 -0734 00 4 00000          PDX 0,4
13414  0500 00 4 00000   E1     CLA 0,4
13415  0601 00 0 02574          STO ECS3
13416  0734 00 4 00000          PAX 0,4
13417  0500 00 4 00000          CLA 0,4
13420  0601 00 0 02575          STO ECS4
13421  0734 00 4 00000          PAX 0,4
13422 -0754 00 4 00000          PXD 0,4
13423  0074 00 4 13703          TSX $EVAL,4
13424  0560 00 0 02573          LDQ ECS2
13425  0100 00 0 13435          TZE E2
13426 -0534 00 4 02575          LXD ECS4,4
13427  0500 00 4 00000          CLA 0,4
13430  0734 00 4 00000          PAX 0,4
13431 -0754 00 4 00000          PXD 0,4
13432  0074 00 4 01736   E3     TSX UNSAVE,4
13433 -0534 00 4 02572          LXD ECS1,4
13434  0020 00 0 13703          TRA $EVAL
13435 -0534 00 4 02574   E2     LXD ECS3,4
13436  3 00000 4 13414          TXH E1,4,0
13437 -0754 00 0 00000          ZAC                   IF OUT OF CLAUSES, VALUE IS NIL
13440  0020 00 0 13432          TRA      E3           ..
                                    BASIC LISP FUNCTIONS FOR APPLY


                         R      HED
                                    CAR

13441  0634 00 4 13446   CARP   SXA      CARX,4
13442 -0734 00 4 00000          PDX ,4
13443  0500 00 4 00000          CLA ,4
13444  0734 00 4 00000          PAX ,4
13445 -0754 00 4 00000          PXD ,4
13446  0774 00 4 00000   CARX   AXT      **,4
13447  0020 00 4 00001          TRA 1,4
13450  0 00000 0 00000   BFS1
```

APPLY AND EVAL

```
13451  0634 00 4 13455  CDRP   SXA     CDRX,4
13452 -0734 00 4 00000          PDX ,4
13453  0500 00 4 00000          CLA ,4
13454 -0320 00 0 00133          ANA BFDM
13455  0774 00 4 00000  CDRX   AXT     **,4
13456  0020 00 4 00001          TRA 1,4
                00133   BFDM   SYN     $DMASK

13457  0634 00 4 13470  ATOMP  SXA     ATMX,4
13460  0100 00 0 13465          TZE ATP1
13461 -0734 00 4 00000          PDX ,4
13462  0500 00 4 00000          CLA ,4
13463  0734 00 4 00000          PAX ,4
13464 -3 77776 4 13467          TXL *+3,4,-2
13465  0500 00 0 00126  ATP1   CLA BFQ1
13466  0020 00 0 13470          TRA *+2
13467 -0754 00 0 00000          PXD ,0
13470  0774 00 4 00000  ATMX   AXT     **,4
13471  0020 00 4 00001          TRA 1,4
                00126   BFQ1   SYN     $QD1

13472  0100 00 0 13475  NULLP  TZE *+3
13473 -0754 00 0 00000          PXD ,0
13474  0020 00 4 00001          TRA 1,4
13475  0500 00 0 00126          CLA BFQ1
13476  0020 00 4 00001          TRA 1,4
```

LAMBDA FOR FUNCTIONAL ARGUMENTS

```
13477 -0634 00 4 13450  LAMP   SXD BFS1,4
13500  0601 00 0 02606          STO BFS2                L
13501  0131 00 0 00000          XCA
13502  0560 00 0 15244          LDQ =0
13503  0074 00 4 02714          TSX $CONS,4             CONS(A,0)
13504  0131 00 0 00000          XCA
13505  0500 00 0 02606          CLA BFS2
13506  0074 00 4 07426          TSX APPEND,4
13507  0131 00 0 00000          XCA
13510  0500 00 0 00145          CLA BFFAG
13511 -0534 00 4 13450          LXD BFS1,4
13512  0020 00 0 02714          TRA $CONS               LIST(FUNARG,L,A)
                00145   BFFAG  SYN     FNARGD
```

LABEL FSUBR

```
13513 -0634 00 4 13450  LABP   SXD BFS1,4
13514 -0600 00 0 02607          STQ BFS3                A
13515 -0734 00 4 00000          PDX ,4                  L
13516  0500 00 4 00000          CLA ,4
13517  0601 00 0 02606          STO BFS2                CWR(L)
13520 -0734 00 4 00000          PDX ,4                  CDR(L)
13521  0500 00 4 00000          CLA ,4
```

APPLY AND EVAL

```
13522  0734 00 4 00000          PAX ,4                    CADR(L)
13523 -0754 00 4 00000          PXD ,4
13524  0601 00 0 02605          STO BFS4
13525  0131 00 0 00000          XCA
13526  0534 00 4 02606          LXA BFS2,4                CAR(L)
13527  0131 00 0 00000          XCA
13530 -0754 00 4 00000          PXD ,4
13531  0074 00 4 02714          TSX $CONS,4               LIST(CAR(L),CADR(L))
13532  0560 00 0 02607          LDQ BFS3
13533  0074 00 4 02714          TSX $CONS,4               CONS(LIST,A)
13534  0131 00 0 00000          XCA
13535  0500 00 0 02605          CLA BFS4                  CADR(L)
13536 -0534 00 4 13450          LXD BFS1,4
13537  0020 00 0 13703          TRA $EVAL
```

SETQ

```
13540 -0634 00 4 02650   SETQP  SXD REPS1,4
13541  0074 00 4 01723          TSX $SAVE,4
13542 -3 02653 0 01766          TXL     $END2,,REPV+2
13543 -0734 00 4 00000          PDX ,4                    L
13544  0500 00 4 00000          CLA ,4
13545  0734 00 4 00000          PAX ,4                    CAR(L)
13546 -0634 00 4 02651          SXD REPV,4
13547 -0734 00 4 00000          PDX ,4                    CDR(L)
13550  0500 00 4 00000          CLA ,4
13551  0734 00 4 00000          PAX ,4                    CADR(L)
13552 -0754 00 4 00000          PXD ,4
13553  0074 00 4 13703          TSX $EVAL,4               EVAL(CADR(L),A)
13554  0601 00 0 02652          STO REPT1
13555  0500 00 0 13570          CLA REPP1
13556  0601 00 0 02531          STO $ARG3
13557  0560 00 0 02526          LDQ     $ALIST          ****CHANGE*******
13560  0500 00 0 02651          CLA REPV
13561  0074 00 4 07663          TSX SASSOC,4              SASSOC(CAR(L),PV,ERROR)
13562 -0734 00 4 00000          PDX ,4
13563  0500 00 0 02652          CLA REPT1
13564  0622 00 4 00000          STD     0,4               REPLACE DECREMENT
13565  0074 00 4 01736          TSX UNSAVE,4
13566 -0534 00 4 02650          LXD REPS1,4
13567  0020 00 4 00001          TRA 1,4

13570 -3 00000 0 13571   REPP1 TXL *+1,,0
13571  0500 00 0 02651          CLA REPV
13572                           ERROR   SETQNV          SETQ GIVEN ON NON-EXISTENT VARIABLE
```

SET

```
13575 -0634 00 4 13450   SETP   SXD BFS1,4
13576  0601 00 0 13616          STO BFS5
13577 -0600 00 0 02606          STQ BFS2
```

APPLY AND EVAL

```
13600  0560 00 0 13611          LDQ SETP1
13601 -0600 00 0 02531          STQ $ARG3
13602  0560 00 0 02636          LDQ PRGVAR
13603  0074 00 4 07663          TSX SASSOC,4
13604 -0734 00 4 00000          PDX ,4
13605  0500 00 0 02606          CLA BFS2
13606  0622 00 4 00000          STD 0,4
13607 -0534 00 4 13450          LXD BFS1,4
13610  0020 00 4 00001          TRA 1,4

13611 -3 00000 0 13612   SETP1 TXL *+1,,0
13612  0500 00 0 13616          CLA BFS5
13613                           ERROR   SETNVR        SET GIVEN ON NON-EXISTENT VARIABLE
13616  0 00000 0 00000   BFS5

                         *              AND SPECIAL FORM
13617 -0100 00 0 13622   EVA8  TNZ EVA6
13620  0500 00 0 00126          CLA EVCT
13621  0020 00 4 00001          TRA 1,4
13622 -0634 00 4 02553   EVA6  SXD EVA1,4
13623  0074 00 4 01723          TSX $SAVE,4
13624 -3 02557 0 01764          TXL     $END3,,EVA9+2      SAVE 3 ITEMS
13625 -0734 00 4 00000          PDX ,4
13626  0500 00 4 00000   EVA4  CLA ,4
13627  0601 00 0 02554          STO EVA2
13630  0734 00 4 00000          PAX ,4
13631 -0754 00 4 00000          PXD ,4
13632 -0600 00 0 02555          STQ EVA9
13633  0074 00 4 13703          TSX $EVAL,4
13634  0560 00 0 02555          LDQ EVA9
13635 -0100 00 0 13641          TNZ EVA3
13636  0074 00 4 01736   EVA5  TSX UNSAVE,4
13637 -0534 00 4 02553          LXD EVA1,4
13640  0020 00 4 00001          TRA 1,4
13641 -0534 00 4 02554   EVA3  LXD EVA2,4
13642  3 00000 4 13626          TXH EVA4,4,0
13643  0500 00 0 00126          CLA EVCT
13644  0020 00 0 13636          TRA EVA5
                         *                OR SPECIAL FORM
13645 -0100 00 0 13650   EVR8  TNZ EVR6
13646 -0754 00 0 00000          ZAC
13647  0020 00 4 00001          TRA 1,4
13650 -0634 00 4 02626   EVR6  SXD EVR1,4
13651  0074 00 4 01723          TSX $SAVE,4
13652 -3 02632 0 01764          TXL     $END3,,EVR9+2      SAVE 3 ITEMS
13653 -0734 00 4 00000          PDX ,4
13654  0500 00 4 00000   EVR4  CLA ,4
13655  0601 00 0 02627          STO EVR2
13656  0734 00 4 00000          PAX ,4
13657 -0754 00 4 00000          PXD ,4
13660 -0600 00 0 02630          STQ EVR9
13661  0074 00 4 13703          TSX $EVAL,4
13662  0560 00 0 02630          LDQ EVR9
13663  0100 00 0 13670          TZE EVR3
13664  0500 00 0 00126          CLA EVCT
```

```
13665  0074 00 4 01736   EVR5  TSX UNSAVE,4
13666 -0534 00 4 02626         LXD EVR1,4
13667  0020 00 4 00001         TRA 1,4
13670 -0534 00 4 02627   EVR3  LXD EVR2,4
13671  3 00000 4 13654         TXH EVR4,4,0
13672 -0754 00 0 00000         ZAC
13673  0020 00 0 13665         TRA EVR5
                    00126 EVCT SYN     $QD1

13674 -0600 00 0 13450   EQP   STQ BFS1
13675  0402 00 0 13450         SUB BFS1
13676 -0100 00 0 13701         TNZ *+3
13677  0500 00 0 00126         CLA BFQ1
13700  0020 00 4 00001         TRA 1,4
13701 -0754 00 0 00000         PXD ,0
13702  0020 00 4 00001         TRA 1,4

                               EVAL(E,A)  5/6/59

                          A    HED
13703 -0634 00 4 02576   EVAL  SXD EVS1,4
13704  0100 00 4 00001         TZE 1,4
13705  0601 00 0 14352         STO EVTE              E
13706 -0734 00 4 00000         PDX ,4
13707  0500 00 4 00000         CLA ,4
13710  0625 00 0 14357         STT     EVLNS         SEE IF A NUMBER
13711  0520 00 0 14357         ZET     EVLNS         SKIP IF NOT A NUMBER
13712  0020 00 0 13771         TRA     EV1N          IS A NUMBER(CONSTANT)
13713  0734 00 4 00000         PAX ,4                CAR(E)
13714  3 77776 4 13774         TXH EVP1,4,-2         = - 1
13715 -0634 00 4 14353         SXD EVTAE,4           CAR(E)
13716  0634 00 4 02576         SXA     EVS1,4        SAVE FUNCTION WITH INDEX REGISTER
13717  0622 00 0 02602         STD EVTDE             CDR(E)
13720  0500 00 4 00000         CLA ,4
13721  0625 00 0 14357         STT     EVLNS         SEE IF A NUMBER
13722  0520 00 0 14357         ZET     EVLNS         TEST FOR A NUMBER
13723  0020 00 0 14240         TRA     EVP26         UNDEFINED FUNCTION IF A NUMBER
13724  0634 00 0 02601         SXA     EVTRK,0
13725 -0634 00 0 02601         SXD     EVTRK,0       ZERO EVTRK (LEAVE MINUS SIGN)
13726  0734 00 4 00000         PAX     0,4
13727 -3 77776 4 14112         TXL     EV27,4,-2
                          *
                          *              CAAR(E) = -1
                          *
13730 -0734 00 4 00000   EVP2  PDX ,4                CDAR(E)
13731 -3 00000 4 14071         TXL EVP25,4,0         NULL(J)
13732  0500 00 4 00000         CLA ,4
13733  0734 00 4 00000         PAX ,4                CAR(J)
13734  3 10405 4 13736         TXH     *+2,4,$TRACE
13735  3 10404 4 13767         TXH     EVTRT,4,$TRACE-1   =TRACE
13736  3 10465 4 13740         TXH *+2,4,$SUBR
13737  3 10464 4 14207         TXH     EVP27,4,$SUBR-1    OF IF A SUBR
13740  3 12065 4 13742         TXH *+2,4,$FSUBR
13741  3 12064 4 14042         TXH EVP22,4,$FSUBR-1       =FSUBR
13742  3 12247 4 13744         TXH *+2,4,$EXPR
```

```
13743   3 12246 4 14063          TXH EVP23,4,$EXPR-1          =EXPR
13744   3 12230 4 13730          TXH EVP2,4,$FEXPR
13745  -3 12227 4 13730          TXL EVP2,4,$FEXPR-1          /= FEXPR
13746   0622 00 0 02603          STD EVD2                     CDR(J)
13747  -0600 00 0 02531          STQ $ARG3                    A
13750   0500 00 0 02531          CLA $ARG3
13751   0560 00 0 15244          LDQ =0        0
13752   0074 00 4 02714          TSX $CONS,4                  CONS(A,0)
13753   0131 00 0 00000          XCA
13754   0500 00 0 02602          CLA EVTDE
13755   0074 00 4 02714          TSX $CONS,4                  LIST(CDR(E),A)
13756   0131 00 0 00000          XCA
13757  -0534 00 4 02603          LXD EVD2,4                   CDR(J)
13760   0500 00 4 00000          CLA ,4
13761   0734 00 4 00000          PAX ,4                       CADR(J)
13762  -0754 00 4 00000          PXD ,4
13763   0520 00 0 02601          ZET     EVTRK                TEST FOR TRACE MODE
13764   0020 00 0 14262          TRA     EVTXP
13765  -0534 00 4 02576          LXD EVS1,4
13766   0020 00 0 13110          TRA $APPLY                   APPLY(CADR(J),LIST(CDR(E),A),A)
                                *
13767  -0625 00 0 02601   EVTRT STL     EVTRK                SET THE TRACE SWITCH
13770   0020 00 0 13730          TRA     EVP2                 GO SEARCH MORE
                                *
                                *
                                *            CAR(E) = -1
                                *
13771   0500 00 0 14352   EV1N  CLA     EVTE                 GET THE NUMBER
13772  -0534 00 4 02576          LXD     EVS1,4               RESTORE LINK INDEX
13773   0020 00 4 00001          TRA     1,4
                                *
13774  -0734 00 4 00000   EVP1  PDX ,4                        J
13775  -3 00000 4 14012          TXL EVP11,4,0                = 0
13776   0500 00 4 00000          CLA ,4
13777   0734 00 4 00000          PAX ,4                       CAR(J)
14000   3 13231 4 13774          TXH EVP1,4,$APVAL            = APVAL
14001  -3 13230 4 13774          TXL EVP1,4,$APVAL-1
14002  -0734 00 4 00000          PDX ,4                       CDR(J)
14003   0500 00 4 00000          CLA ,4
14004   0734 00 4 00000          PAX ,4                       CADR(J)
14005   0500 00 4 00000          CLA ,4
14006   0734 00 4 00000          PAX ,4                       CAADR(J)
14007  -0754 00 4 00000          PXD ,4
14010  -0534 00 4 02576          LXD EVS1,4
14011   0020 00 4 00001          TRA 1,4
                                *
14012  -0600 00 0 14354   EVP11 STQ EVTA                     A
14013   0500 00 0 14352          CLA EVTE                     E
14014   0622 00 0 14027          STD EVI1
14015   0402 00 0 00126          SUB EVQD1
14016   0622 00 0 14030          STD EVI2
14017  -0634 00 2 14356          SXD EVD1,2
14020  -0534 00 4 14354          LXD EVTA,4
14021  -3 00000 4 14036   EVL1  TXL EVP12,4,0                 NULL(J)
14022   0500 00 4 00000      •   CLA ,4
```

```
14023  0734 00 2 00000          PAX ,2                    CAR(J)
14024 -0734 00 4 00000          PDX ,4                    CDR(J)
14025  0500 00 2 00000          CLA ,2
14026  0734 00 2 00000          PAX ,2                    CAAR(J)
14027  3 00000 2 14021   EVI1   TXH EVL1,2,**             CAAR(J) = E
14030 -3 00000 2 14021   EVI2   TXL EVL1,2,**
14031 -0734 00 4 00000          PDX ,4                    CDAR(J)
14032 -0754 00 4 00000          PXD ,4
14033 -0534 00 2 14356          LXD EVD1,2
14034 -0534 00 4 02576          LXD EVS1,4
14035  0020 00 4 00001          TRA 1,4
                        *
14036  0500 00 0 14352  EVP12   CLA     EVTE
14037                           ERROR   UBNDVR            UNBOUND VARIABLE MENTIONED TO EVAL
                        *
14042 -0734 00 4 00000   EVP22  PDX ,4                    CDR(J)     FSUBR
14043  0500 00 4 00000          CLA ,4
14044  0734 00 4 00000          PAX ,4                    CADR(J)
14045  0500 00 4 00000          CLA ,4                    CWADR(J)
14046  0601 00 0 14355          STO EVT1
14047  0500 00 0 02576          CLA     EVS1              ATOM AND IR4 FOR SAVING $ALIST
14050  0601 00 0 02525          STO     CSV
14051  0074 00 4 01723          TSX     $SAVE,4
14052 -3 02530 0 01766          TXL     $END2,,$ALIST+2
14053 -0600 00 0 02526          STQ     $ALIST
14054  0520 00 0 02601          ZET     EVTRK             TEST WHETERT TO TRACT
14055  0020 00 0 14245          TRA     EVTFS             YES,TRACE FSUBR
14056  0500 00 0 02602          CLA     EVTDE             GET BACK ARGUMENTS
14057  0074 00 4 14355          TSX     EVT1,4
14060  0074 00 4 01736          TSX     UNSAVE,4
14061 -0534 00 4 02525          LXD     CSV,4
14062  0020 00 4 00001          TRA     1,4
                        *
                        *               EVP23 THE EXPR BRANCH FOR EVAL
                        *
14063 -0734 00 4 00000   EVP23  PDX     0,4               REST OF PROPERTY LIST
14064  0500 00 4 00000          CLA     0,4               GET THE EXPR
14065  0734 00 4 00000          PAX     0,4
14066 -0634 00 4 14353          SXD     EVTAE,4           SAVE IN TEMPORARY STORAGE
14067 -0534 00 4 01727          LXD     $CPPI,4           PUSH DOWN COUNTER
14070  1 77773 4 14114          TXI     EVP28,4,-5        SAVE 5 ITEMS
                        *
14071  0500 00 0 14353  EVP25  CLA EVTAE                  CAR(E)
14072  0622 00 0 14106          STD EVI3                  TXH
14073  0402 00 0 00126          SUB EVQD1
14074  0622 00 0 14107          STD EVI4                  TXL
14075 -0634 00 2 14355          SXD EVT1,2
14076 -0600 00 0 14356          STQ EVD1
14077 -0534 00 4 14356          LXD EVD1,4                A
14100 -3 00000 4 14240   EVL2   TXL EVP26,4,0             NULL(J)
14101  0500 00 4 00000          CLA ,4
14102 -0734 00 4 00000          PDX ,4                    CDR(J)
14103  0734 00 2 00000          PAX ,2                    CAR(J)
14104  0500 00 2 00000          CLA ,2
14105  0734 00 2 00000          PAX ,2                    CAAR(J)
```

```
14106   3 00000 2 14100   EVI3  TXH EVL2,2,**          /= CAR(E)
14107  -3 00000 2 14100   EVI4  TXL EVL2,2,**
14110  -0534 00 2 14355         LXD     EVTI,2
14111   0622 00 0 14353         STD     EVTAE          SAVE FUNCTION
14112  -0534 00 4 01727   EV27  LXD     $CPPI,4
14113   1 77773 4 14114         TXI     *+1,4,-5       SAVE TOTAL OF 5 ITEMS
14114   0522 00 0 02003   EVP28 XEC     ENDPDL         TEST FOR OUT OF PUSH DOWN LIST
14115  -0634 00 4 01727         SXD     $CPPI,4
14116  -0534 00 1 01727         LXD     $CPPI,1        MAKE IXI AND $CPPI AGREE
14117   0500 00 0 02576         CLA     EVS1
14120   0601 00 4 77773         STO     -5,4
14121   0500 00 0 02577         CLA     EVSE
14122   0601 00 4 77774         STO     -4,4
14123   0500 00 0 02600         CLA     EVSA
14124   0601 00 4 77775         STO     -3,4
14125   0500 00 0 02601         CLA     EVTRK
14126   0601 00 4 77776         STO     -2,4
14127   0500 00 0 14360         CLA     EVCM
14130   0601 00 4 77777         STO     -1,4
14131   0500 00 0 14353         CLA     EVTAE          GET THE FUNCTION
14132   0622 00 0 02577         STD     EVSE
14133  -0600 00 0 02600         STQ EVSA               A
14134   0500 00 0 02602         CLA EVTDE              CDR(E)
14135   0560 00 0 14214         LDQ     ELP1           FUNCTIONAL ARGUMENT
14136   0074 00 4 03201         TSX     MAPLIS,4       MAPLIST(L,EVAL(CAR(L),A))
14137   0601 00 0 14355         STO EVT1
14140   0500 00 0 02600         CLA     EVSA
14141   0601 00 0 02531         STO     $ARG3
14142   0500 00 0 02577         CLA     EVSE
14143  -0534 00 4 01727         LXD     $CPPI,4        START OPEN UNSAVE
14144   0560 00 4 77773         LDQ     -5,4
14145  -0600 00 0 02576         STQ     EVS1
14146   0560 00 4 77774         LDQ     -4,4
14147  -0600 00 0 02577         STQ     EVSE
14150   0560 00 4 77775         LDQ     -3,4
14151  -0600 00 0 02600         STQ     EVSA
14152   0560 00 4 77776         LDQ     -2,4
14153  -0600 00 0 02601         STQ     EVTRK
14154   1 00005 4 14155         TXI     *+1,4,5
14155  -0634 00 4 01727         SXD     $CPPI,4
14156  -0534 00 1 01727         LXD     $CPPI,1        MAKE IXI AND $CPPI AGREE
14157   0560 00 0 14355         LDQ     EVT1
14160   0520 00 0 02601         ZET     EVTRK          TEST RACE SWITCH
14161   0020 00 0 14164         TRA     EVDCO          DECODE EVTRK
14162  -0534 00 4 02576  EVAPG  LXD     EVS1,4
14163   0020 00 0 13110         TRA $APPLY             APPLY(CADAR(J),EVLIS(CDR(E),A),A)
                        *
                        *              IF CAR E IS A SUBR, THE POINTER TO THE TXL INSTRUCTION
                        *              IS SAVED IN THE DECREMENT OF EVTRK.  THE ADDRESS OF
                        *              EVTRK IS THE TRACE SWITCH.
                        *
14164  -0534 00 4 02601  EVDCO  LXD     EVTRK,4        LOOK FOR SUBR POINTER
14165  -3 00000 4 14262         TXL     EVTXP,4,0      THERE ISNT ANY, SO GO AND TRACE EXPR
14166   0534 00 4 02601         LXA     EVTRK,4        SEE IF THE SUBR IS TRACED
14167   3 00000 4 14162         TXH     EVAPG,4,0      YES IT IS. LET APPLY HANDLE IT
```

```
14170 -0534 00 4 02601          LXD     EVTRK,4         GET THE TXL SUBR WORD
14171  0500 00 4 00000          CLA     0,4
14172  0601 00 0 14355          STO     EVT1            READY TO EXECUTE
14173  0500 00 0 02576          CLA     EVS1            GET RETURN INDEX AND ATOM NAME
14174  0601 00 0 02525          STO     CSV             AND SAVE THEM ALONG WITH $ALIST
14175  0074 00 4 01723          TSX     $SAVE,4
14176 -3 02530 0 01766          TXL     $END2,,$ALIST+2
14177  0500 00 0 02531          CLA     $ARG3
14200  0601 00 0 02526          STO     $ALIST          POST CURRENT ALIST
14201  0131 00 0 00000          XCA                     ARGUMENT LIST TO AC
14202  0074 00 4 07713          TSX     $SPREAD,4       SMEAR IT OUT
14203  0074 00 4 14355          TSX     EVT1,4          EXECUTE SUBR
14204  0074 00 4 01736          TSX     UNSAVE,4        RESTORE ALIST AND IX
14205 -0534 00 4 02525          LXD     CSV,4
14206  0020 00 4 00001          TRA     1,4             AND RETURN
                        *
14207 -0734 00 4 00000  EVP27   PDX     0,4             SUBR BRANCH
14210  0500 00 4 00000          CLA     0,4
14211  0734 00 4 00000          PAX     0,4             POINTER TO TXL WORD
14212 -0634 00 4 02601          SXD     EVTRK,4         TO SAVE POSITION
14213  0020 00 0 14112          TRA     EV27            EVALUATE ARGUMENTS
                        *
14214 -3 00000 0 14215   ELP1   TXL     *+1,,0
14215  0634 00 4 14223          SXA     ELT1,4          SAVE LINK IR
14216 -0734 00 4 00000          PDX     ,4              J
14217  0500 00 4 00000          CLA     ,4
14220  0734 00 4 00000          PAX     ,4
14221 -0754 00 4 00000          PXD     ,4              CAR(J)
14222  0560 00 0 02600          LDQ     EVSA            GET CURRENT A LIST
14223  0774 00 4 00000   ELT1   AXT     **,4            RESTORE LINK IR
14224  0020 00 0 13703          TRA $EVAL
                        *
                        *       EVLIS
                        *
14225 -0634 00 4 02576   EVLIS  SXD     EVS1,4          SAVE LINK IR
14226  0774 00 4 12330          AXT     EVLISL,4        ATOM EVLIS
14227  0634 00 4 02576          SXA     EVS1,4          FOR BACKTRACE
14230  0074 00 4 01723          TSX     $SAVE,4         SAVE EVAL STORAGE
14231 -3 02602 0 01764          TXL     $END3,,EVSA+2
14232 -0600 00 0 02600          STQ     EVSA
14233  0560 00 0 14214          LDQ     ELP1
14234  0074 00 4 03201          TSX     MAPLIS,4
14235  0074 00 4 01736          TSX     UNSAVE,4
14236 -0534 00 4 02576          LXD     EVS1,4
14237  0020 00 4 00001          TRA     1,4
                        *
14240 -0534 00 2 14355  EVP26   LXD     EVT1,2
14241  0500 00 0 14352          CLA EVTE
14242                           ERROR   UNDFCN          FUNCTION OBJECT HAS NO DEFINITION
                        *
14245  0734 00 4 00000  EVTFS   PAX     0,4             ATOM NAME
14246 -0754 00 4 00000          PXD     0,4             TO PRINT POSITION
14247  0560 00 0 02602          LDQ     EVTDE
14250  0074 00 4 14301          TSX     ARGOF,4         PRINT ARGUMENT MESSAGE
14251  0560 00 0 02526          LDQ     $ALIST          RESTORE ALIST AFTER ARGOF
```

```
14252  0500 00 0 02602          CLA     EVTDE           AND ARGUMENT LIST
14253  0074 00 4 14355          TSX     EVT1,4          DO THE FSUBR
14254  0074 00 4 01736          TSX     UNSAVE,4        RESTORE THE IR
14255  0131 00 0 00000          XCA                     VALUE TO MQ
14256  0534 00 4 02525          LXA     CSV,4           GET ATOM NAME FOR VALUE MESSAGE
14257 -0754 00 4 00000          PXD     0,4             TO AC
14260 -0534 00 4 02525          LXD     CSV,4           AND RETURN IR4
14261  0020 00 0 14335          TRA     VALOF           PRINT VALUE MESSAGE
                        *
14262  0622 00 0 02602  EVTXP   STD     EVTDE           SAVE LAMBDA EXPRESSION
14263  0534 00 4 02576          LXA     EVS1,4          GET ATOMIC FUNCTION
14264 -0754 00 4 00000          PXD     0,4             TO PRINT POSITION
14265  0074 00 4 14301          TSX     ARGOF,4         PRINT ARGUMENT MESSAGE
14266  0074 00 4 01723          TSX     $SAVE,4         SAVE THERETURN IX
14267 -3 02600 0 01770          TXL     $END1,,EVS1+2
14270  0560 00 0 14355          LDQ     EVT1            RESTORE THE LIST OF ARGUMENTS
14271  0500 00 0 02602          CLA     EVTDE           AND THE LAMBDA EXPRESSION
14272  0074 00 4 13110          TSX     $APPLY,4        APPLY THE FUNCTION TO ITS ARGS
14273  0074 00 4 01736          TSX     UNSAVE,4
14274  0131 00 0 00000          XCA                     PUT VALUE IN MQ
14275  0534 00 4 02576          LXA     EVS1,4          NAME OF ROUTINE TRACED
14276 -0754 00 4 00000          PXD     0,4             PUT IN AC
14277 -0534 00 4 02576          LXD     EVS1,4          LINK IR
14300  0020 00 0 14335          TRA     VALOF           PRINT VALUE OF STATEMENT
                        *
                        *       ARGOF   PRINTS ARGUMENTS OF  NAME FOLLOWED BY THE LIST OF ARGUMEN
                        *
14301  0634 00 4 14325  ARGOF   SXA     PRX,4           SAVE INDEX REGISTERS
14302  0634 00 2 14324          SXA     PRY,2
14303  0601 00 0 14327          STO     AGA             SAVE ATOM NAME
14304 -0600 00 0 14330          STQ     AGQ             SAVE LIST OF ARGUMENTS
14305  0074 00 4 04230          TSX     TERPRI,4        PRINT A BLANK LINE
14306  0774 00 2 00003          AXT     3,2             PRINT2 OUT 3 WORDS
14307  0500 00 2 14334          CLA     AGM+3,2
14310  0074 00 4 04076          TSX     $PRIN2,4
14311  2 00001 2 14307          TIX     *-2,2,1         LOOP
14312  0500 00 0 14327          CLA     AGA
14313  0074 00 4 03564          TSX     $PRINT,4        PRINT OUT THE LINE
14314 -0534 00 2 14330          LXD     AGQ,2           START THE PRINLIS
14315 -3 00000 2 14324  PLL     TXL     PRY,2,0         EXIT IF END OF LIST
14316  0500 00 2 00000          CLA     0,2             NEXT ITEM
14317 -0734 00 2 00000          PDX     0,2             CDR OF LIST
14320  0734 00 4 00000          PAX     0,4             CAR
14321 -0754 00 4 00000          PXD     0,4
14322  0074 00 4 03564          TSX     $PRINT,4
14323  0020 00 0 14315          TRA     PLL             GET NEXT ITEM
14324  0774 00 2 00000  PRY     AXT     **,2            RESTORE INDEX REGISTERS
14325  0774 00 4 00000  PRX     AXT     **,4
14326  0020 00 4 00001          TRA     1,4             EXIT
                        *
14327  0 00000 0 00000  AGA                             TEMPORARY STORAGE
14330  0 00000 0 00000  AGQ
14331  215127644425     AGM     BCI     1,ARGUME
14332 -056362607777             OCT     456362607777    ARGUMENTS
14333 -062660777777     AGO     OCT     462660777777    OF
```

```
14334  652143642560        VALV  BCI     1,VALUE
                           *
                           *       VALOF   PRINTS VALUE OF NAME FOLLOWED BY ONE LIST
                           *               SHARES STORAGE WITH ARGOF ROUTINE
                           *
14335  0634 00 4 14350     VALOF SXA     VAX,4           SAVE LINK IR
14336  0601 00 0 14327           STO     AGA             ATOM NAME
14337 -0600 00 0 14330           STQ     AGQ             VALUE OF EXPRESSION
14340  0074 00 4 04230           TSX     TERPRI,4        PRINT A BLANK LINE
14341  0500 00 0 14334           CLA     VALV            WORD VALUE
14342  0074 00 4 04076           TSX     $PRIN2,4        PUT IN OUTPUT LINE
14343  0500 00 0 14333           CLA     AGO             WORD OF
14344  0074 00 4 04076           TSX     $PRIN2,4
14345  0500 00 0 14327           CLA     AGA             ATOM
14346  0074 00 4 03564           TSX     $PRINT,4        PRINT OUT THE LINE
14347  0500 00 0 14330           CLA     AGQ             VALUE
14350  0774 00 4 00000     VAX   AXT     **,4            RESTORE LINK IR
14351  0020 00 0 03564           TRA     $PRINT          PRINT OUT VALUE AND RETURN
14352  0 00000 0 00000     EVTE                          E
14353  0 00000 0 00000     EVTAE                         CAR(E)
14354  0 00000 0 00000     EVTA                          A
14355  0 00000 0 00000     EVTI
14356  0 00000 0 00000     EVDI
14357  0 00000 0 00000     EVLNS                         TEST CELL FOR NUMBERS
14360 -3 02603 0 01762     EVCM  TXL     $END4,,EVTRK+2
                    00126  EVQD1 SYN     $QD1
                           * INTER       MULTIPLE LISP STATEMENT PROGRAM FEATURE INTERPRETER
                           *             RECODED TO MAKE THE INTERPRETER AND COMPILER PROGRAM
                           *             FEATURE UNDER STAND THE SAME LANGUAGE
                           *
                           R     HED
14361 -0634 00 4 02633     INTER SXD     INTRX,4         SAVE LINK IR
14362  0074 00 4 01723           TSX     $SAVE,4         SAVE PROTECTED TEMPCRARY STORAGE
14363 -3 02641 0 01760           TXL     $END5,,INTGS+2  SAVE 5 ITEMS
14364  0634 00 2 02635           SXA     INTGL,2           SAVE  INDEX REGISTER 2
14365 -0600 00 0 02636           STQ     INTPL           SAVE PAIR LIST
14366  0600 00 0 02637           STZ     INTGS           ZERO THE GO SWITCH
14367 -0734 00 4 00000           PDX     0,4             POINTER TO PROGRAM
14370  0500 00 4 00000           CLA     0,4             FIRST WORD
14371  0622 00 0 02634           STD     INTB            POINTER TO BEGINNING OF PROGRAM
14372  0622 00 0 14522           STD     INTE            DITTO
14373  0734 00 4 00000           PAX     0,4             POINTER TO LIST OF PROGRAM VARIBLES
14374 -0754 00 4 00000           PXD     0,4             TO DECREMENT
14375  0560 00 0 14473           LDQ     INTFB           FUNCTIONAL ARGUMENT
14376  0074 00 4 03201           TSX     MAPLIS,4        (MAPLIST PV (LAMBDA (L) (CONS (CAR L)
14377  0560 00 0 02636           LDQ     INTPL           NIL)))   PICK UP PAIR LIST
14400  0074 00 4 07562           TSX     $NCONC,4        ATTACH PROGARM VARIBLES TO PAIR LIST
14401  0601 00 0 02636           STO     INTPL           PUT IN PAIR LISDT REGISTER
14402  0560 00 0 15244           LDQ     =0              ZERO THE MQ
14403 -0534 00 4 14522     INTGM LXD     INTE,4          SEARCH PROGRAM FOR GO TO POINTS
14404 -3 00000 4 14417           TXL     INTAA,4,0       GO IF END OF PROGRAM
14405  0500 00 4 00000           CLA     0,4             NEXT WORD
14406  0622 00 0 14522           STD     INTE            SAVE CDR
14407  0734 00 2 00000           PAX     0,2             CAR
14410  0500 00 2 00000           CLA     0,2             MAKE ATOM TEST
```

APPLY AND EVAL

```
14411  0734 00 2 00000          PAX     0,2
14412 -3 77776 2 14403          TXL     INTGM,2,-2      GO IF NOT AN ATOM
14413 -0754 00 4 00000          PXD     0,4             IS AN ATOM, PUT POINTER TO CURRENT LOC
14414  0074 00 4 02714          TSX     $CONS,4         PUT ON GO LCST
14415  0131 00 0 00000          XCA                     ANSWER TO MQ
14416  0020 00 0 14403          TRA     INTGM           NEXT ITEM
14417 -0620 00 0 02635  INTAA   SLQ     INTGL           ALL DONE, STORE GO LIST
14420 -0534 00 4 02634  INTGA   LXD     INTB,4,0        NEXT PROGRAM LOCATION
14421 -3 00000 4 14513          TXL     INTRN,4,0       RETURN WITH NIL IF RAN OUT OF STATEMEN
14422  0500 00 4 00000          CLA     0,4             NEXT WORD
14423  0622 00 0 02634          STD     INTB            SAVE CDR
14424  0734 00 4 00000          PAX     0,4             CAR
14425  0500 00 4 00000          CLA     0,4             FIRST WORD
14426  0734 00 2 00000          PAX     0,2             CHECK FOR ATOM OR $COND
14427  3 77776 2 14420          TXH     INTGA,2,-2      GO TO NEXT STEP IF ATOM
14430 -3 12646 2 14452          TXL     INTEV,2,$COND-1 GO TO EVAL IF NOT $COND
14431  3 12647 2 14452          TXH     INTEV,2,$COND
14432 -0734 00 2 00000          PDX     0,2             IS $COND DO AN EVCOND
14433 -3 00000 2 14420  INTEB   TXL     INTGA,2,0       GO TO NEXT STEP IF COND UNSATISFIED
14434  0500 00 2 00000          CLA     0,2             FIRST COND STATEMENT
14435 -0734 00 2 00000          PDX     0,2             CDR
14436  0734 00 4 00000          PAX     0,4             FIRST SUB COND
14437  0500 00 4 00000          CLA     0,4
14440 -0734 00 4 00000          PDX     0,4             POINTER TO THEN PART
14441  0634 00 4 02634          SXA     INTB,4          SAVE IN PROTECTED STORAGE
14442  0734 00 4 00000          PAX     0,4             POINTER TO IF PART
14443 -0754 00 4 00000          PXD     0,4             PUT IN DECREMENT
14444  0560 00 0 02636          LDQ     INTPL           PAIR LIST
14445  0074 00 4 13703          TSX     $EVAL,4         EVALUATE IT
14446  0100 00 0 14433          TZE     INTEB           GO IF IF PART IS FALSE
14447  0534 00 4 02634          LXA     INTB,4          GET THEN PART
14450  0500 00 4 00000          CLA     0,4
14451  0734 00 4 00000          PAX     0,4             PPINTER TPO THEN PART
14452 -0754 00 4 00000  INTEV   PXD     0,4             LIST TO BE EVALUATED
14453  0560 00 0 02636          LDQ     INTPL           GET PAIR LIST
14454  0074 00 4 13703          TSX     $EVAL,4         EVALUATE IT
14455 -0520 00 0 02637          NZT     INTGS           SEE IF GO SWITCH SET
14456  0020 00 0 14420          TRA     INTGA           GO TO NEXT STATEMENT
14457  0534 00 4 02637          LXA     INTGS,4         WAS SET, SEE IF GO OR RETURN
14460  3 77776 4 14513          TXH     INTRN,4,-2      TRA IF RETURN
14461 -0754 00 4 00000          PXD     0,4             POINTER TO ITEM
14462  0560 00 0 14504          LDQ     INTFC           GET SASSOC FUNCTIONAL ARGUMENT
14463 -0600 00 0 02531          STQ     $ARG3           PUT IN $ARG3
14464  0560 00 0 02635          LDQ     INTGL           GET GO LIST
14465  0074 00 4 07663          TSX     SASSOC,4        SEARCH FOR ATOM
14466 -0734 00 4 00000          PDX     0,4             POINTER TP  PROGRAM POINT
14467  0500 00 4 00000          CLA     0,4             TAKE CDR
14470  0622 00 0 02634          STD     INTB            SET PROGRAM POINT
14471  0600 00 0 02637          STZ     INTGS           ZERO THE GO SWITCH
14472  0020 00 0 14420          TRA     INTGA           GO TO THAT STATEMENT
                        *
14473 -3 00001 0 14474  INTFB   TXL     *+1,,1          MAPLIST FUNCTIONAL ARGUMENT
14474  0634 00 4 14502          SXA     INTFX,4         (LAMBDA (L) (CONS (CAR L) NIL))
14475 -0734 00 4 00000          PDX     0,4
14476  0500 00 4 00000          CLA     0,4
```

```
14477  0734 00 4 00000          PAX     0,4
14500 -0754 00 4 00000          PXD     0,4
14501  0560 00 0 15244          LDQ     =0
14502  0774 00 4 00000   INTFX  AXT     **,4
14503  0020 00 0 02714          TRA     $CONS
                         *
14504 -3 00001 0 14505   INTFC  TXL     *+1,,1          UNLABELED GO TO POINT ERROR
14505  0534 00 2 02637          LXA     INTGS,2         POINTER TO GO POINT LABEL
14506 -0754 00 2 00000          PXD     0,2             PUT IN DECREMENT
14507  0534 00 2 02635          LXA     INTGL,2         RESTORE INDEX REGISTER 2
14510                            ERROR   UNDFGO      GO-TO POINT NOT LABELED
                         *
14513 -0534 00 4 02637   INTRN  LXD     INTGS,4         RETURN VALUE
14514 -0754 00 4 00000          PXD     0,4             PUT IN DECREMENT
14515  0600 00 0 02637          STZ     INTGS           ZERO THE GO SWITCH
14516  0534 00 2 02635          LXA     INTGL,2         RESTORE INDEX REGISTER 2
14517  0074 00 4 01736          TSX     UNSAVE,4        RESTORE PROTECTED STORAGE
14520 -0534 00 4 02633          LXD     INTRX,4         RESTORE LINK IR
14521  0020 00 4 00001          TRA     1,4
                         *               TEMPORARY STORAGE FOR INTERPRETER
14522  0 00000 0 00000   INTE                           TEMPORARY STORAGE
                  02636  PRGVAR SYN     INTPL
                         *
                         *
                         * RETURN                      SPECIAL PROGRAM SETS RETURN SWITCH
                         *                             IN PROGRAM INTERPRETER
                         *
14523 -0501 00 0 00132   RETURN ORA     $AMASK          SIGNAL THAT IT IS A RETURN
14524  0601 00 0 02637          STO     INTGS           SET UP GO SWITCH
14525  0500 00 0 00126          CLA     $QD1            PICK UP TRUTH VALUE
14526  0020 00 4 00001          TRA     1,4             EXIT
                         *
                         * GO                          SPECIAL FORM FOR PROGRAM INTERPRETER, GIVES GO TO POINT
                         *
14527 -0634 00 4 02604   GOGOGO SXD     GOX,4           SAVE LINK IR
14530 -0734 00 4 00000          PDX     0,4             POINTER TO ARGUMENT LIST
14531  0500 00 4 00000          CLA     0,4
14532  0621 00 0 02637          STA     INTGS           PUT CAR IN GO SWITCH
14533  0734 00 4 00000          PAX     0,4             CAR TO IR
14534  0500 00 4 00000          CLA     0,4             GET FIRST WORD
14535  0734 00 4 00000          PAX     0,4             SEE IF ATOMIC
14536  3 77776 4 14547          TXH     GOT,4,-2        EXIT TRUE IF ATIMIC
14537  0534 00 4 02637          LXA     INTGS,4         OTHERWISE GET ARGUMENT
14540 -0754 00 4 00000          PXD     0,4             PUT INDECREMENT
14541  0074 00 4 01723          TSX     $SAVE,4         SAVE LINK IR
14542 -3 02606 0 01770          TXL     $END1,,GOX+2    SAVE 1 ITEM
14543  0074 00 4 13703          TSX     $EVAL,4         EVALUATE THE ARGUMENT
14544  0074 00 4 01736          TSX     UNSAVE,4        RSTORE LINK IR
14545 -0734 00 4 00000          PDX     0,4             VALUE
14546  0634 00 4 02637          SXA     INTGS,4         PU IN GO SWITCH
14547  0500 00 0 00126   GOT    CLA     $QD1            TRUTH VALUE
14550 -0534 00 4 02604          LXD     GOX,4           RESTORE LINK IR
14551  0020 00 4 00001          TRA     1,4             EXIT
```

```
                                    HEAD    C                       COMPILER PERMANENT SUBROUTINES
                                    *       LINK HANDLES ALL SUBROUTINE CALLS FROM COMPILED FUNCTION
                                    *       IT REPLACES STR WITH TSX IF SUBROUTINE BEING CALLED
                                    *       IS A SUBR OR FSUBR
                                    *       IT GOES TO APPLY IF THE CALL IS TO EXPR OR FEXPR WITH
                                    *       $ALIST AS THIRD ARGUMENT
                                    *       LINK EXPECTS A TAG OF 7 IN THE STR INST, NAME OF FUNCTION
                                    *       IN THE ADDRESS, AND THE NUMBER OF ARGUMENTS IN THE DECREM
                                    *       ENT                     LINK WILL GO TO THE ROUTINE WHICH
                                    *       HANDLES ERROR TRAPS IF THE CALLING INST DOESNT HAVE A 7
                                    *       TAG
                                    *
14552  0601 00 0 02610  LINK    STO     LNKA
14553 -0600 00 0 02611           STQ     LNKB            SAVE AC AND MQ
14554  0634 00 4 14740           SXA     LER,4           SAVE IR4
14555  0535 00 4 00000           LAC     0,4             COMP POINTER TO STR+1
14556  1 00001 4 14557           TXI     *+1,4,1         MAKE ORDINARY TSX POINTER
14557  0500 00 4 00000           CLA     0,4             GET STR INST  7
14560  0601 00 0 14752           STO     LNKD            SAVE IT
14561 -0320 00 0 00137           ANA     TAGMSK          CHECK FOR 7 TAG
14562  0322 00 0 00137           ERA     TAGMSK
14563 -0100 00 0 14740           TNZ     LER             IF NOT 7 TAG
14564 -0634 00 4 14751           SXD     LNKC,4          SAVE POINTER
14565  0500 00 0 00076           CLA     B$ZERC          RESTORE NIL
14566  0601 00 0 00000           STO     0
14567  0600 00 0 14745           STZ     LNTRS           RESET TRACE SWITCH
14570  0534 00 4 14752           LXA     LNKD,4          FUNCTION ATOM
14571  0500 00 4 00000           CLA     0,4             START PROPERTY LIST SEARCH
14572  0734 00 4 00000           PAX     ,4              IS REALLY ATOM
14573 -3 77776 4 14627           TXL     LNNF,4,-2
14574 -0734 00 4 00000  LNLP     PDX     0,4
14575 -3 00000 4 14627           TXL     LNNF,4,0        NO DEFINITION SO FN VARIABLE
14576  0500 00 4 00000           CLA     0,4
14577  0734 00 4 00000           PAX     0,4
14600 -3 10464 4 14602           TXL     *+2,4,$SUBR-1
14601 -3 10465 4 14666           TXL     LNSBR,4,$SUBR
14602 -3 12064 4 14604           TXL     *+2,4,$FSUBR-1
14603 -3 12065 4 14666           TXL     LNSBR,4,$FSUBR
14604 -3 10404 4 14606           TXL     *+2,4,$TRACE-1
14605 -3 10405 4 14631           TXL     LNTR,4,$TRACE
14606 -3 12246 4 14610           TXL     *+2,4,$EXPR-1
14607 -3 12247 4 14612           TXL     LNEXP,4,$EXPR
14610 -3 12227 4 14574           TXL     LNLP,4,$FEXPR-1
14611  3 12230 4 14574           TXH     LNLP,4,$FEXPR
14612 -0734 00 4 00000  LNEXP    PDX     0,4             EXPR-FEXPR BRANCH
14613  0500 00 4 00000           CLA     0,4
14614  0734 00 4 00000           PAX     0,4             LAMBDA EXPRESSION
14615 -0634 00 4 14746  LNGN     SXD     LNFN,4          SAVE IT
14616  0520 00 0 14745           ZET     LNTRS           TRACE TEST
14617  0020 00 0 14646           TRA     LNTEX           TRACE EXPR OF FEXPR
14620  0074 00 4 14703           TSX     LNARS,4         LIST ARGUMENTS
14621  0131 00 0 00000           XCA
14622  0500 00 0 02526           CLA     $ALIST
14623  0601 00 0 02531           STO     $ARG3           PROPER ALIST
14624  0500 00 0 14746           CLA     LNFN            LAMBDA EXPRESSION
```

```
14625 -0534 00 4 14751        LXD     LNKC,4          RETURN IR
14626  0020 00 0 13110        TRA     $APPLY          DO
                      *
14627  0534 00 4 14752 LNNF   LXA     LNKD,4          FUNCTION DEFN IS ON ALIST
14630  0020 00 0 14615        TRA     LNGN            APPLY WILL TAKE CARE OF THIS
                      *
14631 -0625 00 0 14745 LNTR   STL     LNTRS           SET TRACE SWITCH
14632  0601 00 0 14747        STO     LNAC            SAVE AC
14633  0074 00 4 14703        TSX     LNARS,4         LIST ARGUMENTS
14634  0601 00 0 14750        STO     LNRGL           AND SAVE THEM
14635  0131 00 0 00000        XCA                     TO PRINTING POSITION
14636  0534 00 4 14752        LXA     LNKD,4          ATOM NAME
14637  0634 00 4 14751        SXA     LNKC,4          SAVE WITH INDEX REGISTER
14640 -0754 00 4 00000        PXD     0,4             ALSO FOR TRACE MESSAGE
14641  0074 00 4 01723        TSX     $SAVE,4         SAVE NAME AND RETRN
14642 -3 14753 0 01770        TXL     $END1,,LNKC+2
14643  0074 00 4 14301        TSX     A$ARGOF,4       PRINT ARGUMENTS
14644  0500 00 0 14747        CLA     LNAC            RESTORE AC
14645  0020 00 0 14574        TRA     LNLP            AND CONTINUE PROPERTY LIST SEARCH
                      *
14646  0500 00 0 14746 LNTEX  CLA     LNFN            TRACE EXPR OR FEXPR
14647  0560 00 0 02526        LDQ     $ALIST
14650 -0600 00 0 02531        STQ     $ARG3
14651  0560 00 0 14750        LDQ     LNRGL           SET UP ARGUMENTS OF APPLY
14652  0074 00 4 13110        TSX     $APPLY,4        AND DO THE FUNCTION
14653  0074 00 4 01736 LNTEN  TSX     UNSAVE,4        GET BACK IR4 AND FN NAME
14654  0131 00 0 00000        XCA
14655  0534 00 4 14751        LXA     LNKC,4          ATOM NAME TO AC
14656 -0754 00 4 00000        PXD     0,4
14657 -0534 00 4 14751        LXD     LNKC,4          RESTORE INDEX
14660  0020 00 0 14335        TRA     A$VALOF         PRINT VALUE MESSAGE
                      *
14661  0621 00 0 14664 LNTSB  STA     LNDIS           TRACE SUBR OF FSUBR
14662  0500 00 0 02610        CLA     LNKA            RESTORE AC
14663  0560 00 0 02611        LDQ     LNKB            AND MQ
14664  0074 00 4 00000 LNDIS  TSX     **,4            EXECUTER SUBROUTINE
14665  0020 00 0 14653        TRA     LNTEN           AND REPORT VALUE
                      *
14666 -0734 00 4 00000 LNSBR  PDX     0,4             SUBR OR FSUBR BRANCH
14667  0500 00 4 00000        CLA     0,4
14670  0734 00 4 00000        PAX     0,4
14671  0500 00 4 00000        CLA     0,4             TXL SUBR,,N WORD
14672  0520 00 0 14745        ZET     LNTRS           TEST FOR TRACING
14673  0020 00 0 14661        TRA     LNTSB
14674  0621 00 0 14753        STA     LNTSX           MAKE A TSX
14675  0500 00 0 14753        CLA     LNTSX           GET IT
14676 -0534 00 4 14751        LXD     LNKC,4          RETURN IQ
14677  0601 00 4 00000        STO     0,4             CHANGE THE STR TO TSX
14700  0500 00 0 02610        CLA     LNKA            RESTORE AC
14701  0560 00 0 02611        LDQ     LNKB
14702  0020 00 4 00000        TRA     0,4             GO TO NEW TSX
                      *
14703  0634 00 4 14734 LNARS  SXA     LNLX,4          SUBROUTINE WHICH LISTS ARGS
14704 -0534 00 4 14752        LXD     LNKD,4          NUMBER OF ARGS
14705 -3 00000 4 14736        TXL     LNN,4,0         LST WONT WORK ON ZERO THINGS
```

```
14706 -0634 00 4 14710          SXD     LNKP,4          PUT IN LST ARG POSITICN
14707  0074 00 4 14754          TSX     LST,4           LIST THEM
14710  3 00000 0 02610  LNKP    TXH     LNKA,0,**
14711  0734 00 0 02611          PAX     LNKB,0
14712  0734 00 0 02531          PAX     $ARG3,0
14713  0734 00 0 02532          PAX     $ARG4,0
14714  0734 00 0 02533          PAX     $ARG5,0
14715  0734 00 0 02534          PAX     $ARG6,0
14716  0734 00 0 02535          PAX     $ARG7,0
14717  0734 00 0 02536          PAX     $ARG8,0
14720  0734 00 0 02537          PAX     $ARG9,0
14721  0734 00 0 02540          PAX     $ARG10,0
14722  0734 00 0 02541          PAX     $ARG11,0
14723  0734 00 0 02542          PAX     $ARG12,0
14724  0734 00 0 02543          PAX     $ARG13,0
14725  0734 00 0 02544          PAX     $ARG14,0
14726  0734 00 0 02545          PAX     $ARG15,0
14727  0734 00 0 02546          PAX     $ARG16,0
14730  0734 00 0 02547          PAX     $ARG17,0
14731  0734 00 0 02550          PAX     $ARG18,0
14732  0734 00 0 02551          PAX     $ARG19,0
14733  0734 00 0 02552          PAX     $ARG20,0
14734  0774 00 4 00000  LNLX    AXT     **,4            RESTORE INTEX
14735  0020 00 4 00001          TRA     1,4
14736 -0754 00 0 00000  LNN     PXD     0,0             NIL
14737  0020 00 0 14734          TRA     LNLX
                        *
14740  0774 00 4 00000  LER     AXT     **,4                    RESTORE IR4
14741  0500 00 0 02610          CLA     LNKA
                        ***CHANGE HERE NECESSITATED BY DELETION OF STRPNT******
14742                    ERROR   STRTRP
                        *               LINK STORAGE
                        *               IS HERE, EXCEPT FOR LINKA NAD LINKB WHICH ARE IN GARBAG
14745  0 00000 0 00000  LNTRS                          TRACE SWITCH
14746  0 00000 0 00000  LNFN                           FUNCTION DEFINITICN
14747  0 00000 0 00000  LNAC                           TEMPORARY AC STORAGE
14750  0 00000 0 00000  LNRGL                          ARGS LISTED DURNING TRACE INTERLUDE
14751  0 00000 0 00000  LNKC                           IX4 POINTER TC STR WORD
14752  0 00000 0 00000  LNKD                           CONTAINS STR NAME,7,NUM
14753  0074 00 4 00000  LNTSX   TSX     **,4            INSTRUCTION TC BE PLANTED
                        *               LST IS THE SUBROUTINE WHICH DOES LISTING IN COMPILED
                        *               FUNCTIONS       N ELEMENTS WHERE N IS IN AC, ARE
                        *               LISTED          ARGUMENTS ARE GOTTEN BY CLA*
                        *               FROM THE N REGISTERS SUCCEEDING THE CALL
                        *
14754  0634 00 2 15015  LST     SXA     LX2,2           SAVE IR2
14755  0500 00 4 00001          CLA     1,4             TO GET N FROM FIRSTDECREMENT
14756  0622 00 0 14760          STD     LSN             TO DECREMENT IR4 FOR POINT EXIT
14757  0622 00 0 14774          STD     LSC             TO DECREMENT THE CONS COUNTER
14760 -2 00000 4 14761  LSN     TNX     *+1,4,**
14761 -0734 00 2 00000          PDX     0,2             N TO IR2
14762 -0754 00 4 00000          PXD     0,4             START TO COMPLEMENT IR4
14763 -0737 00 4 00000          PDC     0,4             OH FOR A 7094
14764  1 00001 4 14765          TXI     *+1,4,1         ONE MORE FOR EXIT
```

```
14765   0634 00 4 15001          SXA     LSP,4           SET UP GET INST
14766   0634 00 4 15017          SXA     LSE,4           AND RETURN
14767  -0534 00 4 02735          LXD     $FREE,4         FIRST FREE WORD
14770   3 00000 4 14772          TXH     *+2,4,0         TEST FOR OUT OF FREE
14771   0074 00 4 03037          TSX     $FROUT,4        WILL RETURN -2,4
14772  -0634 00 4 15026          SXD     LAN,4           THE ANSWER TO THIS SAUSAGE CONS
14773   0534 00 4 02726          LXA     $CNTR1,4        GET CONS COUNTER
14774   2 00000 4 14777  LSC     TIX     *+3,4,**        REDUCE IT BY N
14775   0074 00 4 02736          TSX     ARREST,4        OUT OF CONSES
14776   0774 00 4 77777          AXT     -1,4            RESET COUNTER (UP TO N CONSES MAY BE
14777   0634 00 4 02726          SXA     $CNTR1,4        LOST EVERY 7777 OCTAL CONSES)
15000  -0534 00 4 15026          LXD     LAN,4           RESTORE  IR4 TO FREE WORD POINTER
15001   0500 60 2 00000  LSP     CLA*    **,2            GET ARGUMENT
15002   0771 00 0 00022          ARS     18              TO ADDRESS
15003   0621 00 4 00000          STA     0,4             PUT IT IN THE FREE WORD ADDR
15004   0500 00 4 00000          CLA     0,4             NEXT  FREE WORD
15005   0634 00 4 15013          SXA     LFX,4           SAVE PRECEDING WORD TO CUT OFF
15006  -0734 00 4 00000  LSR     PDX     0,4             NEXT FREE WORD TO IR
15007  -3 00000 4 15020          TXL     LFIX,4,0        OUT OF FREE STORAGE?
15010   2 00001 2 15001          TIX     LSP,2,1         COUNT DOWN
15011   0622 00 0 02735          STD     $FREE           RESTORE FREE
15012  -0754 00 0 00000          PXD     0,0             CLEAR
15013   0774 00 4 00000  LFX     AXT     **,4            LAST WORD IN LIST
15014   0622 00 4 00000          STD     0,4             GETS NIL IN ITS DECREMENT
15015   0774 00 2 00000  LX2     AXT     **,2            RESTORE IR2
15016   0500 00 0 15026          CLA     LAN             GET THE ANSWER
15017   0020 00 0 00000  LSE     TRA     **              RETURN
15020   0500 00 0 15026  LFIX    CLA     LAN             TO GET IT PROTECTED DURING MOP UP
15021   0074 00 4 02007          TSX     RECLAM,4
15022   0500 00 0 02735          CLA     $FREE           FIX UP THE SAUSAGE
15023   0522 00 0 15013          XEC     LFX             GET LAST WORD TO IR
15024   0622 00 4 00000          STD     0,4             FIX ITS DECREMENT
15025   0020 00 0 15006          TRA     LSR
15026   0 00000 0 00000  LAN     PZE
                         *       UNWND IS UNSAVE FOR COMPILED FUNCTIONS,USED BY ERRORSET
                         *       TO RESTORE THE PDL TO PRISTINE STATE
15027   0634 00 4 15062  UNWND   SXA     UNX,4
15030  -0600 00 0 15065          STQ     UNSQ            SAVE THE MQ FOR BACKTRACE
15031  -0534 00 1 01727          LXD     $CPPI,1
15032  -0754 00 1 00000          PXD     0,1
15033   0401 00 1 77777          ADM     -1,1
15034   0402 00 0 00126          SUB     $QD1
15035   0622 00 0 15037          STD     UNTL
15036   1 00001 1 15037  UNLP    TXI     *+1,1,1
15037   3 00000 1 15061  UNTL    TXH     UNF,1,**
15040   0500 00 1 00000          CLA     0,1
15041   0734 00 4 00000          PAX     0,4
15042  -3 00000 4 15036          TXL     UNLP,4,0
15043  -0754 00 4 00000          PXD     0,4
15044   0131 00 0 00000          XCA
15045   0500 00 4 00000          CLA     0,4
15046   0734 00 4 00000          PAX     0,4
15047   3 77776 4 15036          TXH     UNLP,4,-2
15050   0500 00 0 15176          CLA     PROS
15051   0520 00 0 01712          ZET     BPLACE
```

```
15052  0500 00 0 15177          CLA     PROSCM        NONZERO IMPLIES COMPILER
15053  0074 00 4 10547          TSX     GET,4
15054  0100 00 0 15036          TZE     UNLP
15055  0500 00 1 00000          CLA     0,1
15056  0734 00 4 00000          PAX     0,4
15057  0622 00 4 00000          STD     0,4
15060  0020 00 0 15036          TRA     UNLP
15061 -0634 00 1 01727  UNF     SXD     $CPPI,1
15062  0774 00 4 00000  UNX     AXT     **,4
15063  0560 00 0 15065          LDQ     UNSQ
15064  0020 00 4 00001          TRA     1,4
15065  0 00000 0 00000  UNSQ                                   SAVE MQ HERE
                        *   MOVE IS A SPECIAL COMPILER SERVICE SUBROUTINE WITH BAD CALLING.
                        *       TSX     *MOVE,1
                        *       TNX     NAME,1,*MN
15066  0634 00 1 15073  MOVE    SXA     MOVY,1
15067 -0534 00 1 01727          LXD     $CPPI,1                PICK UP PDL PPINTER
15070  0601 00 1 00001          STO     1,1                    SAVE AC
15071 -0600 00 1 00002          STQ     2,1
15072 -0634 00 4 15162          SXD     TXLW,4                 SAVE RETURN INDEX
15073  0774 00 4 00000  MOVY    AXT     **,4                   PICK UP REFERENCE TO CALLING HEAD   .
15074  0500 00 4 00001          CLA     1,4                    TNX WORD HAS NAME IN ADDR.
15075  0621 00 0 15162          STA     TXLW                   COMPLETES THE TXL WORD
15076  0622 00 0 15161          STD     STRW                   PUT N IN STRW DECREMENT
15077  0500 00 0 15162          CLA     TXLW
15100  0601 00 1 00000          STO     0,1                    PUT IT AT HEAD OF PDL BLOCK
15101  0500 00 4 00000          CLA     0,4                    TSX HAS COUNT FIELD
15102 -0320 00 0 00136          ANA     CNTMSK                 COUNT FIELD MASK
15103  0100 00 0 15153          TZE     MOVD                   IF LESS THAN 3 ARGS
15104 -0734 00 4 00000          PDX     0,4                    COUNT FIELD TO IX
15105  0020 00 4 15152          TRA     MOVD-1,4               ENTER PART OF MOVE ROUTINE

                        MOVEMC MACRO    NUMBER        CHAIN OF CLA $ARGX      STO  X,1
                               IRP      NUMBER
                               CLA      $ARG'NUMBER   HERE IS THE PROMISED CLA
                               STO      NUMBER,1      AND HERE IS THE STO
                               IRP
                        MOVEMC END

15106                          MOVEMC   (20,19,18,17,16,15,14,13,12,11)
15132                          MOVEMC   (10,9,8,7,6,5,4,3)

15152  0534 00 4 15073          LXA     MOVY,4                 RESTORE IR4
15153  0522 00 4 00001  MOVD    XEC     1,4          XECED TNX DECREMENTS IX1 BUT NO TRANSFER
15154 -0634 00 1 01727          SXD     $CPPI,1                KEEP CPPI UP TO DATE ALSO
15155  0500 00 0 15161          CLA     STRW                   CREATES SECOND PARAMETER WORD
15156  0601 00 1 77777          STO     -1,1                   PUT AT VERY END OF BLOCK
15157 -3 00000 1 02004  ENPDL1  TXL     $NOPDL,1,**    ***CHANGE********
15160  0020 00 4 00002          TRA     2,4                    RETURN FROM LINK
15161 -1 00000 0 00000  STRW    STR     **
15162 -3 00000 0 00000  TXLW    TXL     **,,**
                        *
                        * RESTOR PICKS UP IX4 FROM PDL,SETS BACK CPPI ,AND EXITS.
15163 -0634 00 1 01727  RESTOR SXD     $CPPI,1
15164  0131 00 0 00000          XCA                            SAVE VALUE OF FUNCTION
```


49196

```
15165  0500 00 1 00000          CLA     0,1          PICK UP RETURN WORD
15166 -0734 00 4 00000          PDX     0,4          RESTORE IX4
15167  0131 00 0 00000          XCA                  RESTORE AC
15170  0020 00 4 00001          TRA     1,4          EXIT
                        *
                 15171  PROBE   SYN     *            BEGINNING OF PROTECTED AREA
15171  0 00000 0 00000  LIST                         MAIN LISTING GOES HERE
15172  0 00000 0 00000  QTLST                        THE LIST OF QUOTES.NEVER ERASE
15173  0 00000 0 00000  QTLSTC                  COMPILERS LIST OF QUOTES ERASED BY EXCISE
15174  0 00000 0 00000  TAB                          TEMPORARY SYM TABLE
15175  0 00000 0 00000  LCOM                         STORAGE FOR COMMON ONLY.PROTECTED
15176  0 00000 0 00000  PROS                         PROTECTED FUNCTION NAMES AND SPECIALS
15177  0 00000 0 00000  PROSCM                  COMPILERS SPECIALS,ERASED BY EXCISE
                 15177  PROEN   SYN     *-1          END OF PROTECTED AREA
15200  0 00000 0 15066  MOV             MOVE         THE WORD POINTED TO BY SYM ON *MOVE
15201  0 00000 0 14754  LSTR            LST          POINTED TO BY SYM ON ATOM *LIST
15202  0 00000 0 15163  RTRN            RESTOR
15203  0 00000 0 02526    ALST          $ALIST
                 15204  LITRLS EQU     *
```

PERMANENT LIST STRUCTURE

```
                    64260          ORG     26800          COMPILER MUST FIT IN REM. SPACE
64260               LOWERP BSS     1                  LWER LIMIT OF PERMENANT LIST STRUCTURE
                                   **********************HEAD OR HED**********************************
                    0      HED

                                LOWER LIMIT OF PERM. LIST STRUCTURE

                    BUCK5  MACRO                      GENERATE 5 MEMBERS OF THE BUCKET
                                   ,,-*+1
                                   ,,-*+1
                                   ,,-*+1
                                   ,,-*+1
                                   ,,-*+1             MAN, OH MAN, WHAT AN UGLY MACRO
                    BUCK5  END

                    BUCK25 MACRO                      GUESS WHAT THIS ONE DOES
                           BUCK5
                           BUCK5
                           BUCK5
                           BUCK5
                           BUCK5
                    BUCK25 END

64261  0 00000 0 00000         PZE     ,,             LAST BUCKET
64262                          BUCK25                 NOW WE HAVE 26
64313                          BUCK25                 51
64344                          BUCK25                 76
64375                          BUCK25                 101
64426                          BUCK25                 126
64457  0 13322 0 00000  BUCKET                ,,-*+1  POINTER TO BUCKETS
                         64457 OBLIST SYN      BUCKET
```

PERMANENT LIST STRUCTURE

```
               INSERT  $ATOMS
        ATOM   MACRO   PN,PROPS,NAME PRINT NAME LETTER BY LETTER
        NAME   SET     -*
                       -1,,-*-1          ATOM HEADER.  POINTS TO P-LIST
               IRP     PROPS             FOR EACH NON-PNAME PROPERTY
               PROPTY  ('PROPS')         ASSEMBLE SOME LIST STRUCTURE
               IRP                       ..
                       PNAME,,-*-1       FLAG
                       -*-1              END OF PLIST.  PTR TO FW LIST
               PNAME   ('PN')
               ADDOB   NAME
        ATOM   END

        PROPTY MACRO   PROP,LINK         LINK IS /CRS/ TO NEXT PROPERTY
               PROP,LINK                 PROP SHOULD BE READY TO ACCEPT A FINAL ARG
        LINK   SET     *                 LINK GOES ON, AS PROMISED, TO NEXT PROP
        PROPTY END

        *              HERE ARE SOME SAMPLE PROPERTIES

        APVAL  MACRO   VALUE,LINK        (APVAL,VALUE)
                       APVAL,,-*-1       FLAG
                       -*-1,,-LINK       POINTER TO POINTER TO VALUE
                       VALUE,,0          POINTER TO VALUE
        APVAL  END

        FSUBR  MACRO   ENTRY,NARGS,LINK
                       FSUBR,,-*-1       FLAG
                       -*-1,,-LINK       POINTER TO TXL
               TXL     ENTRY,,NARGS
        FSUBR  END                       THIS ONE WAS JUST LIKE SUBR

        NAPVAL MACRO   NUMBER,LINK,F1 NUMERICAL APVAL
               APVAL   -F1,LINK
        F1     MZE     -1,1,-NUMBER      LITTLE NUMBER ATOM JUST FOR HERE
        NAPVAL END

        SYM    MACRO   LOC,LINK          (SYM,LOC)
                       SYM,,-*-1         FLAG
                       -LOC,,-LINK       POINTER TO LOC
        SYM    END

        SUBR   MACRO   ENTRY,NARGS,LINK (SUBR,ENTRY,NARGS)
                       $SUBR,,-*-1       FLAG
                       -*-1,,-LINK       POINTER TO TXL
               TXL     ENTRY,,NARGS      TXL TO MACHINE CODE
        SUBR   END
```

PERMANENT LIST STRUCTURE

```
*               PNAME HAS THE HAIRY TASK OF COUNTING LETTERS AND SUCH

PNAME   MACRO   LTRS,VFDS       ONLY THE FIRST ARGUMENT IS REAL
N       SET     0               NO LETTERS YET
        IRP     LTRS            ALLOT THE LIST STRUCTURE BY A SIMPLE COUNT
N       SET     N+1             ..
        IRP     ..
NP5     SET     N+5             THIS FOR ROUNDING UPWARD TO FULL WORDS
PWORDS  SET     NP5/6           NUMBER OF BCD WORDS
        LINKS   PWORDS,-VFDS    BUILD THE LIST STRUCTURE FIRST
VFDS    CHOOSE  (2,3,4,5,6,7,8,9,10,11,12,13,14,15),('LTRS')
PNAME   END

CHOOSE  MACRO   M,LTRS          CHOOSE PNAMX TO APPLY TO LTRS
        IRP     M
        IFF     N-M,X           N IS FREE FOR THE LETTER-COUNT
        PNAM'M,LTRS
        IRP
CHOOSE  END

PNAM2   MACRO   A,B
        VFD     H12/A'B,O24/77777777
PNAM2   END

PNAM3   MACRO   A,B,C
        VFD     H18/A'B'C,O18/777777
PNAM3   END

PNAM4   MACRO   A,B,C,D
        VFD     H24/A'B'C'D,O12/7777
PNAM4   END

PNAM5   MACRO   A,B,C,D,E
        VFD     H30/A'B'C'D'E,O6/77
PNAM5   END

PNAM6   MACRO   A,B,C,D,E,F
        BCI     1,A'B'C'D'E'F
PNAM6   END

PNAM7   MACRO   A,B,C,D,E,F,G
        BCI     1,A'B'C'D'E'F
        VFD     H6/G,O30/7777777777
PNAM7   END

PNAM8   MACRO   A,B,C,D,E,F,G,H
        BCI     1,A'B'C'D'E'F
        PNAM2   G,H
PNAM8   END
```

PERMANENT LIST STRUCTURE

```
PNAM9   MACRO   A,B,C,D,E,F,G,H,I
        BCI     1,A'B'C'D'E'F
        PNAM3   G,H,I
PNAM9   END

PNAM10  MACRO   A,B,C,D,E,F,G,H,I,J
        BCI     1,A'B'C'D'E'F
        PNAM4   G,H,I,J
PNAM10  END

PNAM11  MACRO   A,B,C,D,E,F,G,H,I,J,K
        BCI     1,A'B'C'D'E'F
        PNAM5   G,H,I,J,K
PNAM11  END

PNAM12  MACRO   A,B,C,D,E,F,G,H,I,J,K,L
        BCI     1,A'B'C'D'E'F
        BCI     1,G'H'I'J'K'L
PNAM12  END

PNAM13  MACRO   A,B,C,D,E,F,G,H,I,J,K,L,M
        BCI     1,A'B'C'D'E'F
        PNAM7   G,H,I,J,K,L,M
PNAM13  END

PNAM14  MACRO   A,B,C,D,E,F,G,H,I,J,K,L,M,N
        BCI     1,A'B'C'D'E'F
        PNAM8   G,H,I,J,K,L,M,N
PNAM14  END

PNAM15  MACRO   A,B,C,D,E,F,G,H,I,J,K,L,M,N,O
        BCI     1,A'B'C'D'E'F
        PNAM9   G,H,I,J,K,L,M,N,O
PNAM15  END


LINKS   MACRO   NWDS,MVFDS,F1 NWDS LINKS TO FULL WORDS AT VFDS
        IFF     NWDS-1,X      IF THIS IS THE LAST ONE
                MVFDS,,0
        IFF     NWDS-1        OTHERWISE
                MVFDS,,-F1
        IFF     NWDS-1        STILL OTHERWISE
F1      LINKS   NWDS-1,MVFDS-1
LINKS   END

ADDOB   MACRO   OBNAME        ADD THE NAMED OBJECT TO THE OBLIST
        PZE     OBNAME,,OLDOBS OLDOBS IS SET EACH TIME
OLDOBS  SET     -*+1          ..  HERE, FOR EXAMPLE (-(*-1))
ADDOB   END
```

PERMANENT LIST STRUCTURE

```
                 00000  OLDOBS SET      0            ZERO HERE MEANS NIL
        64460                  ATOM     (A,D,D,1),((SUBR,ADD1,1))
        64471                  ATOM     (A,D,V,A,N,C,E),((SUBR,ADVANC,0))
        64504                  ATOM     ($,A,L,I,S,T),((SYM,C$ALST))
        64514                  ATOM     (A,N,D),((FSUBR,$EVAND,0)),AND
        64525                  ATOM     (A,P,P,E,N,D),((SUBR,APPEND,2)),F1
        64536                  ATOM     (A,P,P,L,Y),((SUBR,APPLY,3))
        64547                  ATOM     (A,P,V,A,L),,APVAL
                 13231  APVAL1 SYN      APVAL
        64555                  ATOM     (A,R,R,A,Y),((SUBR,ARYMAK,1)),ARRAY
        64566                  ATOM     (A,T,O,M),((SUBR,ATOMP,1)),ATOM
        64577                  ATOM     (A,T,T,R,I,B),((SUBR,ATTRIB,2))

        64610                  ATOM     (B,A,C,K,T,R,A,C,E),((SUBR,BACKTR,0))
        64623                  ATOM     (B,L,A,N,K),((APVAL,H60))
        64634                  ATOM     (B,P,S,R,E,M,A,I,N,D,E,R),((SUBR,BPSREM,0))

        64647                  ATOM     (C,A,R),((SUBR,CARP,1))
        64660                  ATOM     (C,D,R),((SUBR,CDRP,1))
        64671                  ATOM     (C,A,A,R),((SUBR,CAARXX,1))
        64702                  ATOM     (C,A,D,R),((SUBR,CADRXX,1))
        64713                  ATOM     (C,D,A,R),((SUBR,CDARXX,1))
        64724                  ATOM     (C,D,D,R),((SUBR,CDDRXX,1))
        64735                  ATOM     (C,A,A,A,R),((SUBR,CAAARX,1))
        64746                  ATOM     (C,A,A,D,R),((SUBR,CAADRX,1))
        64757                  ATOM     (C,A,D,A,R),((SUBR,CADARX,1))
        64770                  ATOM     (C,A,D,D,R),((SUBR,CADDRX,1))
        65001                  ATOM     (C,D,A,A,R),((SUBR,CDAARX,1))
        65012                  ATOM     (C,D,A,D,R),((SUBR,CDADRX,1))
        65023                  ATOM     (C,D,D,A,R),((SUBR,CDDARX,1))
        65034                  ATOM     (C,D,D,D,R),((SUBR,CDDDRX,1))
        65045                  ATOM     (C,H,A,R,C,O,U,N,T),((NAPVAL,CHACT))
        65061                  ATOM     (C,L,E,A,R,B,U,F,F),((SUBR,CLEAR,0))
        65074                  ATOM     (C,O,L,O,N),((APVAL,H35))
        65105                  ATOM     (C,O,M,M,A),((APVAL,H73))
        65116                  ATOM     (C,O,M,P,R,I,N,T),((SUBR,CCMPRI,1))
        65131                  ATOM     (C,O,N,D),((FSUBR,$EVCON,0)),COND
        65142                  ATOM     (C,O,N,S),((SUBR,CONS,2))
        65153                  ATOM     (C,P,1),((SUBR,CP1,1))
        65164                  ATOM     (C,O,P,Y),((SUBR,$COPY,1)),COPYN
        65175                  ATOM     (C,O,U,N,T),((SUBR,COUNT,0))
        65206                  ATOM     (C,R),((APVAL,H55))
        65217 0 12560 0 77777  )PJ1   PZE      -1,,-*-1     CURCHAR ATOM COULDN'T BE MACRO'ED
        65220 0 12557 0 13231                  APVAL,,-*-1  FOR ONE THING, THIS APVAL IS ONLY 1 LEVEL
        65221 0 12556 0 66620                  -CURC1,,-*-1 ..
        65222 0 12555 0 11227                  PNAME,,-*-1  HERE IS THE PNAME FLAG
        65223 0 12550 0 12554                  -)PJ1A,,-)PJ1B CAR IS PNAME STRUCTURE, CDR IS REST
        65224                  )PJ1A  PNAME    (C,U,R,C,H,A,R) THE OLD MACROS AREN'T HOPELESS
        65230 0 12547 0 10540  )PJ1B           SPECAL,,-*-1 FOR ANOTHER THING, WE ARE NOT SET FOR SPECL
        65231 0 00000 0 66617                  -CURC
                 12561  PJ1    SYN      -)PJ1
        65232                  ADDOB    PJ1
```

PERMANENT LIST STRUCTURE

```
65233                    ATOM    (D,A,S,H),((APVAL,H40))
65244                    ATOM    (D,I,F,F,E,R,E,N,C,E),((SUBR,DIFFER,2))
65257                    ATOM    (D,I,G,I,T),((SUBR,DIGIT,1))
65270                    ATOM    (D,I,V,I,D,E),((SUBR,DIVIDE,2))
65301                    ATOM    (D,O,L,L,A,R),((APVAL,H53))

65312                    ATOM    (E,N,D,R,E,A,D),((SUBR,ENDRED,0))
65325                    ATOM    (E,O,F),((APVAL,H12))
65336                    ATOM    (E,O,R),((APVAL,H72))
65347                    ATOM    (E,Q),((SUBR,EQP,2))
65360                    ATOM    (E,Q,P),((SUBR,EQPROG,2))
65371                    ATOM    (E,Q,S,I,G,N),((APVAL,H13))
65402                    ATOM    (E,Q,U,A,L),((SUBR,EQUAL,2)),F8
65413                    ATOM    (E,R,R,O,R),((SUBR,ERROR1,1))
65424                    ATOM    (E,R,R,O,R,S,E,T),((SUBR,ERRSET,3)),ERSETO
65437                    ATOM    (E,V,A,L),((SUBR,EVAL,2))
65450                    ATOM    (E,V,L,I,S),((SUBR,$EVLIS,2)),EVLISL
65461                    ATOM    (E,X,C,I,S,A,B,L,E),((SUBR,EXCABL,2))
65474                    ATOM    (E,X,C,I,S),((SUBR,E$EXCIS,1))
65505                    ATOM    (E,X,P,L,O,D,E),((SUBR,EXPLOD,1))
65520                    ATOM    (E,X,P,S,U,B),((SUBR,Q$EXP,1))
65531                    ATOM    (E,X,P,R),,EXPR
65537                    ATOM    (E,X,P,T),((SUBR,EXPT,2))

65550                    ATOM    (F,E,X,P,R),,FEXPR
65556                    ATOM    (F,I,L,E,1),((SUBR,FILE1,4))
65567                    ATOM    (F,I,L,E,D,E,L,E,T,E),((SUBR,FILDEL,2))
65602                    ATOM    (F,I,L,E,E,N,D,R,D),((SUBR,FENDRD,2))
65615                    ATOM    (F,I,L,E,G,O,N,E),((SUBR,NOFILE,2))
65630                    ATOM    (F,I,L,E,S,E,E,K),((SUBR,FILSEK,2))
65643                    ATOM    (F,I,X),,FIX
                 12135   BIN     SYN     FIX
65651                    ATOM    (F,I,X,P),((SUBR,FIXP,1))
65662                    ATOM    (F,L,O,A,T),,FLOAT
65670                    ATOM    (F,L,O,A,T,P),((SUBR,FLOATP,1))
65701                    ATOM    (F,S,L,E,F,T),((NAPVAL,E$FSC))
65713                    ATOM    (F,S,U,B,R),,FSUBR
65721                    ATOM    (F,U,N,A,R,G),,FUNARG
65727                    ATOM    (F,U,N,C,T,I,O,N),((FSUBR,$LAMP,0))
65742                    ATOM    (F,W,L,E,F,T),((NAPVAL,E$FWC))

65754                    ATOM    (G,C,G,A,G),((SUBR,GCGAG,1))
65765                    ATOM    (G,E,N,S,Y,M),((SUBR,GENSYM,0))
65776                    ATOM    (G,E,T),((SUBR,C$GET,2))
66007                    ATOM    (G,E,T,B,C,D),((SUBR,GETBCD,1))
66020                    ATOM    (G,O),((FSUBR,GOGOGO,1)),GO
66031                    ATOM    (G,R,E,A,T,E,R,P),((SUBR,GRTRTP,2))

66044                    ATOM    (I,N,T,E,R,N),((SUBR,INTRN1,1))

66055                    ATOM    (L,A,B,E,L),((FSUBR,LABP,0)),LABEL
66066                    ATOM    (L,A,M,B,D,A),,LAMBDA
66074                    ATOM    (L,A,P),((SUBR,C$LAP,2)),LAP
66105                    ATOM    (L,E,F,T,S,H,I,F,T),((SUBR,LSHIFT,2))
66120                    ATOM    (L,E,S,S,P),((SUBR,LESSTP,2))
```

```
66131   ATOM    (L,I,S,T),((FSUBR,EVLIS,0)),LIST
66142   ATOM    (L,O,G,A,N,D),((FSUBR,LOGAND,0)),PJ37
66153   ATOM    (L,O,G,O,R),((FSUBR,LOGOR,0)),PJ36
66164   ATOM    (L,O,G,S,U,B),((SUBR,Q$LOG,1))
66175   ATOM    (L,O,G,X,O,R),((FSUBR,LOGXOR,0)),PJ38
66206   ATOM    (L,P,A,R),((APVAL,H74))

66217   ATOM    (M,A,P),((SUBR,MAPCAR,2)),PMAPCA
66230   ATOM    (M,A,P,C,O,N),((SUBR,MAPCON,2)),R69B
66241   ATOM    (M,A,P,L,I,S,T),((SUBR,MAPLIS,2)),R69A
66254   ATOM    (M,A,X),((FSUBR,MAX,2)),MAXP
66265   ATOM    (M,I,N),((FSUBR,MIN,2)),MINP
66276   ATOM    (M,I,N,U,S),((SUBR,MNSPRG,1))
66307   ATOM    (M,I,N,U,S,P),((SUBR,MINUSP,1))
66320   ATOM    (M,K,N,A,M),((SUBR,MKNAM,0))
66331   ATOM    (M,K,N,O),((SUBR,MKNO,2))

66342   ATOM    (N,C,O,N,C),((SUBR,NCONC,2))
66353   ADDOB   NIL
66354   ATOM    (N,O,T),((SUBR,NULLP,1))
66365   ATOM    (N,U,L,L),((SUBR,NULLP,1))
66376   ATOM    (N,U,M,B,E,R,P),((SUBR,NUMBRP,1))
66411   ATOM    (N,U,M,O,B),((SUBR,NUMOB,0))
66422   ATOM    (N,U,M,V,A,L),((SUBR,NUMVAL,1))

66433   ATOM    (O,B,L,I,S,T),((APVAL,-OBLIST))
66444   ATOM    (O,C,T,A,L),,OCT
66452   ATOM    (O,N,E,P),((SUBR,ONEP,1))
66463   ATOM    (O,R),((FSUBR,$EVOR,0)),OR

66474   ATOM    (P,A,C,K),((SUBR,PACK,1))
66505   ATOM    (P,A,I,R),((SUBR,PAIR,2))
66516   ATOM    (P,E,R,I,O,D),((APVAL,H33))
66527   ATOM    (P,L,U,S),((FSUBR,ADDP,2)),PLUS
66540   ATOM    (P,L,U,S,S),((APVAL,H20))
66551   ATOM    (P,N,A,M,E),,PNAME
66557   ATOM    (P,R,I,M,E),((APVAL,H14))
66570   ATOM    (P,R,I,N,1),((SUBR,PRIN1,1))
66601   ATOM    (P,R,I,N,T),((SUBR,PRINT,1)),F4
66612   ATOM    (P,R,I,N,T,2),((SUBR,PRINT2,1))
66623   ATOM    (P,R,O,G),((FSUBR,INTER,0)),PROG
66634   ATOM    (P,R,O,P),((SUBR,APROP,3))
66645   ATOM    (P,U,N,C,H),((SUBR,$PUNCH,1))
66656   ATOM    (P,U,N,C,H,T),((SUBR,PUNCHT,0))
66667   ATOM    (P,U,N,C,H,P),((SUBR,PUNCHP,0))

66700   ATOM    (Q,U,O,T,E),((FSUBR,CARP,0)),QUOTE
66711   ATOM    (Q,U,O,T,I,E,N,T),((SUBR,QUOTEN,2))

66724   ATOM    (R,D,F,L,X),((SUBR,RDFLX,0))
66735   ATOM    (R,E,A,D),((SUBR,READ,0)),F13
66746   ATOM    (R,E,C,I,P),((SUBR,RCPPRG,1))
66757   ATOM    (R,E,C,L,A,I,M),((SUBR,RECLAM,0))
66772   ATOM    (R,E,M,A,I,N,D,E,R),((SUBR,REMAIN,2))
67005   ATOM    (R,E,M,P,R,O,P),((SUBR,REMPRP,2))
```

PERMANENT LIST STRUCTURE

```
67020                          ATOM    (R,E,T,U,R,N),((SUBR,RETURN,1))
67031                          ATOM    (R,P,A,R),((APVAL,H34))
67042                          ATOM    (R,P,L,A,C,A),((SUBR,RPLACA,0))
67053                          ATOM    (R,P,L,A,C,D),((SUBR,RPLACD,0))
67064                          ATOM    (R,P,L,A,C,W),((SUBR,RPLACW,0))

67075                          ATOM    (*,L,I,S,T),((SYM,C$LSTR))
67105                          ATOM    (*,M,O,V,E),((SYM,C$MOV))
67115                          ATOM    (*,R,E,T,U,R,N),((SYM,C$RTRN))
67127                          ADDOB   1               *T*

67130                          ATOM    (S,A,S,S,O,C),((SUBR,APSSOC,3))
67141                          ATOM    (S,A,V,B,K),((SUBR,SAVBK,0))
67152                          ATOM    (S,E,A,R,C,H),((SUBR,SEARCH,4)),SRCH
67163                          ATOM    (S,E,T),((SUBR,SETP,2)),SET
67174                          ATOM    (S,E,T,Q),((FSUBR,SETQP,0)),SETQ
67205                          ATOM    (S,E,T,B,K),((SUBR,SETBK,0))
67216                          ATOM    (S,L,A,S,H),((APVAL,H61))
67227                          ATOM    (S,P,E,A,K),((SUBR,SPEAK,0))
67240                          ATOM    (S,P,E,C,I,A,L),,SPECAL
67250                          ATOM    (S,T,A,R),((APVAL,H54))
67261                          ATOM    (S,T,A,R,T,R,E,A,D),((SUBR,STREAD,0))
67274                          ATOM    (S,T,O,P),,STOP
67302                          ATOM    (S,U,B,1),((SUBR,SUB1,1))
67313                          ATOM    (S,U,B,R),,SUBR
67321                          ATOM    (S,U,B,S,T),((SUBR,SUBST,3))
67332                          ATOM    (S,Y,M),,SYM

67340                          ATOM    (T,A,B),((SUBR,TABTAB,1))
67351                          ATOM    (T,E,R,P,R,I),((SUBR,TERPRI,0))
67362                          ATOM    (T,I,M,E,S),((FSUBR,MULT,2)),TIMES
67373                          ATOM    (T,R,A,C,E),,TRACE
67401                          ATOM    (T,Y,T,A,B),((APVAL,H72H)) APVAL IS TAP CHARACTER
67412  0 10365 0 77777  HH72H         -1,,-*-1        HERE IS THAT CHARACTER
67413  0 10364 0 11227                 PNAME,,-*-1     ..
67414  0 00000 0 10363                 -*-1            ..
67415  0 00000 0 10362                 -*-1            ..
67416  727777777777                    VFD     06/72,030/7777777777
                 10366  H72H           SYN     -HH72H
67417                                  ADDOB   H72H

67420                          ATOM    (U,N,C,O,U,N,T),((SUBR,UNCONT,1))
67433                          ATOM    (U,N,P,A,C,K),((SUBR,UNPACK,1))

67444                          ATOM    (Z,E,R,O,P),((SUBR,ZEROP,1))
```

PERMANENT LIST STRUCTURE

```
        *               MACROS TO SPEED DESCRIPTION OF ALPHABETIC OBJECTS

DULL    MACRO   LTR             TAKES OCTAL ARGUMENTS
        IRP     LTR             FOR EACH SUPPLIED ARGUMENT
        DULL1   HH'LTR,LTR      MAKE UP A PNAME STRUCTURE
        IRP
DULL    END

DULL1   MACRO   LOC,OCT         MAKES UP A PNAME STRUCTURE FOR OCT AT LOC
LOC             PNAME,,-*-1     FLAG
                -*-1            POINTER TO LIST OF (ONE) FULL WORD
                -*-1            POINTER TO FULL WORD
        VFD     06/OCT,030/7777777777 PNAME
DULL1   END


ILLEGL  MACRO   LTR             TAKES OCTAL ARGUMENTS
        IRP     LTR             FOR EACH SUPPLIED ARGUMENT
        ILEGL1  HH'LTR,LTR      MAKE UP A PNAME STRUCTURE
        IRP
ILLEGL  END

ILEGL1  MACRO   LOC,OCT         MAKE UP $ILXX$ FOR OCT AT LOC
LOC             PNAME,,-*-1     FLAG
                -*-1            POINTER TO LIST OF (ONE) FULL WORD
                -*-1            LIST OF (ONE) FULL WORD
        VFD     H18/$IL,H12/OCT,H6/$
ILEGL1  END


ADHOBS  MACRO   LTRS            SPECIALIZED MACRO TO PUT H00,H01,... ON OBL
        IRP     LTRS
        ADDOB   H'LTRS          WRITE ADHOBS   (00,01,02, ..., 77)
        IRP
ADHOBS  END


HOBS    MACRO   LTRS            MACRO FOR CREATING ALPHABETIC OBJECTS
        IRP     LTRS            WRITE   HOBS   (77,76,...,00)
        HOB     H'LTRS,LTRS     LOCATION,LTRS FOR REST OF ATOM
        IRP
HOBS    END


HOB     MACRO   NAME,OCT        MACRO FOR ONE ALPH OBJECT
NAME    SYN     -*              DEFINE ATOM WITHOUT )HXX CRAP
                -1,HOBTAG,-HH'OCT ATOM HEADER
HOB     END
```

PERMANENT LIST STRUCTURE

```
                                  *       PROPERTY LISTS FOR ALPHABETIC OBJECTS
                                  *
 67455  0 00000 0 00000   HH00            0
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67456  0 00000 0 00001   HH01            1
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67457  0 00000 0 00002   HH02            2
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67460  0 00000 0 00003   HH03            3
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67461  0 00000 0 00004   HH04            4
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67462  0 00000 0 00005   HH05            5
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67463  0 00000 0 00006   HH06            6
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67464  0 00000 0 00007   HH07            7
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67465 +000000000010      HH10    OCT     10
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67466 +000000000011      HH11    OCT     11
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67467  0 10310 0 11227    HH12   PZE PNAME,,-*-1           END OF FILE
 67470  0 10305 0 10307           PZE -*-1,,-*-3
 67471  0 00000 0 10306           PZE -*-1
 67472 -132546265377              OCT 532546265377          BCD $EOF$
 67473  0 10304 0 13231           PZE APVAL1,,-*-1
 67474  0 00000 0 10303           PZE -*-1
 67475  0 00000 0 07654           PZE H12
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67476                            DULL    (13,14)        =, PRIME
 67506                            ILLEGL  (15,16,17)
 67522                            DULL    (20,21,22,23,24,25) +, A, B, C, D, E
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67552  0 10225 0 77777   HH26        -1,,-*-1              F
 67553  0 10224 0 13231               APVAL,,-*-1
 67554  0 10222 0 10223               -*-1,,-*-2
 67555  0 00000 0 00000               0
 67556  0 10221 0 11227               PNAME,,-*-1
 67557  0 00000 0 10220           PZE -*-1
 67560  0 00000 0 10217           PZE -*-1
 67561 +267777777777              OCT 267777777777
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67562                            DULL    (27,30,31)     G, H, I
 67576                            ILLEGL  32
 67602                            DULL    (33,34,35)     PERIOD, RPAR,COLON
 67616                            ILLEGL  (36,37)
 67626                            DULL    (40,41,42,43,44,45,46,47,50,51) -,J,K,...,R
 67676                            ILLEGL  52
 67702                            DULL    53             $
                                      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 67706  0 10071 0 11227   HH54        PNAME,,-*-1           *
 67707  0 10066 0 10070           PZE     -*-1,,-*-3
 67710  0 00000 0 10067           PZE -*-1
```

PERMANENT LIST STRUCTURE

```
67711 -147777777777          OCT 547777777777
67712  0 10065 0 10446       PZE     SYM,,-*-1
67713  0 00000 0 00054       PZE     -C$STAR
                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
67714                        DULL    55              CR
67720                        ILLEGL  (56,57)
67730                        DULL    (60,61,62)      BLANK, SLASH, S
                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
67744  0 10033 0 77777  HH63      -1,,-*-1                    T
67745  0 10032 0 13231            APVAL,,-*-1
67746  0 10030 0 10031            -*-1,,-*-2
67747  0 00000 0 00001            1
67750  0 10027 0 11227            PNAME,,-*-1
67751  0 00000 0 10026       PZE -*-1
67752  0 00000 0 10025       PZE -*-1
67753 -237777777777          OCT 637777777777
                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
67754                        DULL    (64,65,66,67,70,71) U, V, W, X, Y, Z
                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
70004  0 07773 0 11227  HH72      PNAME,,-*-1                 END OF RECORD
70005  0 07770 0 07772       PZE -*-1,,-*-3
70006  0 00000 0 07771       PZE -*-1
70007 -132546515377          OCT 532546515377            BCD $EUR$
70010  0 07767 0 13231       PZE APVAL1,,-*-1
70011  0 00000 0 07766       PZE -*-1
70012  0 00000 0 07734       PZE H72
                             * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
70013                        DULL    (73,74)         COMMA, LPAR
70023                        ILLEGL  (75,76,77)

                   00000  HOBTAG SET     0                TOP DOWN FOR INDEXING=NON-NUMBERS FIRST
70037                     HOBS    (77,76,75,74,73,72,71,70,67,66,65,64,63,62,61,60)
70057                     HOBS    (57,56,55,54,53,52,51,50,47,46,45,44,43,42,41,40)
70077                     HOBS    (37,36,35,34,33,32,31,30,27,26,25,24,23,22,21,20)
70117                     HOBS    (17,16,15,14,13,12)
                   00001  HOBTAG SET     1                OBJECTS H00,H01,...,H11 ARE NUMBERS
70125                     HOBS    (11,10,07,06,05,04,03,02,01,00)

                   *                NOW LET'S PUT THE STUPID THINGS ON THE OBLIST

70137                     ADHOBS  (00,01,02,03,04,05,06,07,10,11,12,13,14,15,16,17)
70157                     ADHOBS  (20,21,22,23,24,25,26,27,30,31,32,33,34,35,36,37)
70177                     ADHOBS  (40,41,42,43,44,45,46,47,50,51,52,53,54,55,56,57)
70217                     ADHOBS  (60,61,62,63,64,65,66,67,70,71,72,73,74,75,76,77)

70237  0 07542 0 77777  OBLB      -1,,OLDOBS      THIS IS THE OBLIST, WHICH MUST BE LAST
                   70240  UPERML SYN     OBLB+1
```

PERMANENT LIST STRUCTURE

		INSERT	$SYNS
		HEAD	0
07654	EOF	SYN	H12
12626	ADD1	SYN	Q$ADD1
12353	ADDP	SYN	Q$ADDP
13243	APP2	SYN	A$APP2
13110	APPLY	SYN	A$APPLY
07632	APROP	SYN	R$PROP
13457	ATOMP	SYN	R$ATOMP
13441	CARP	SYN	R$CARP
13451	CDRP	SYN	R$CDRP
11162	CHACT	SYN	F$CHACT
10753	CLEAR	SYN	F$CLEAR
03334	COPY	SYN	R$COPY
07317	CP1	SYN	C$CP1
11161	CURC	SYN	F$CURC
11160	CURC1	SYN	F$CURC1
03076	DECON	SYN	E$DECON
11037	DIGIT	SYN	F$DIGIT
13674	EQP	SYN	R$EQP
03450	EQUAL	SYN	L$EQUAL
13703	EVAL	SYN	A$EVAL
13617	EVAND	SYN	R$EVA8
13406	EVCON	SYN	A$EVCON
14225	EVLIS	SYN	A$EVLIS
13645	EVOR	SYN	R$EVR8
12060	EXPT	SYN	Q$EXPT
12713	FIXP	SYN	Q$FIXP
14361	INTER	SYN	R$INTER
13513	LABP	SYN	R$LABP
13477	LAMP	SYN	R$LAMP
11207	LOGOR	SYN	H$LOGOR
12375	MAX	SYN	Q$MAX
12367	MIN	SYN	Q$MIN
10721	MKNAM	SYN	F$MKNAM
11163	MKNO	SYN	F$MKNO
12361	MULT	SYN	Q$MULT
07562	NCONC	SYN	R$NCONC
00000	NIL	SYN	0
13472	NULLP	SYN	R$NULLP
06607	NUMBR	SYN	F$NUMBR
10647	NUMOB	SYN	F$NUMOB
06531	NUTRN	SYN	T$NUTRN
12760	ONEP	SYN	Q$ONEP
10610	PACK	SYN	F$PACK
07447	PAIR	SYN	A$PAIR
03577	PRIN0	SYN	T$PRIN0
03665	PRIN1	SYN	T$PRIN1
04076	PRIN2	SYN	T$PRIN2
03564	PRINT	SYN	T$PRINT
07632	PROP	SYN	R$PROP
04344	PUN2	SYN	T$PUN2
04327	PUNCH	SYN	T$PUNCH
05047	RD	SYN	I$RD
04755	READ	SYN	I$READ

PERMANENT LIST STRUCTURE

```
          13575    SETP   SYN R$SETP
          13540    SETQP  SYN      R$SETQP
          12655    SUB1   SYN      Q$SUB1
          07350    SUBST  SYN R$SUBST
          13012    UNFIX  SYN      Q$UNFIX
          06521    VALUE  SYN I$VALUE
          12734    ZEROP  SYN      Q$ZEROP
          07663    APSSOC SYN SASSOC
70240                     BSS      1-*          FOOL FAP ABOUT LENGTH OF P L STRUCTURE
          76626           ORG      32150        **LETS HOPE IT FITS**
```

```
                                * SETUP         TAKES  SIZE PARAMETERS AND SETS UP THE DEPENDENT CELLS
                                *               MAINLY IN THE RECLAIMER (GARBAGGE COLLECTOR) AND STRPNT
                                *
                                        HEAD    E
                                *
                                *       RESETP  ALTERNATE ENTRANCE TO SETUP TO CHANGE COMPOSITION OF
                                * FREE STRORAGE SLIGHTLY.
                                *
                                *
                                *       EXCISE (NIL) THROWS OUT ALL RPOGRAMS IN EXCISABLE SPACE.
                                *       THIS IS INTENDED FOR USE BY THE COMPILER, BUT IT MAY BE USED IN
                                *               OTHER WAYS AT THE USER'S RISK.
                                *       THE SPACE THUS THROWN AWAY IS CONVERTED TO FREE STORAGE, BUT
                                *               NO ATTEMPT IS MADE TO DISCONNECT THE NAMES OF THE
                                *               PROGRAMS THUS THROWN OUT, SO ANY FURTHER USE OF THESE
                                *               WILL RESULT IN OBSCURITY.
                                *       EXCISE (T) THROWS OUT LAP AS WELL.
                                *
 76626  0634 00 4 77042  EXCIS  SXA     SUPX,4          SAVE IR 4
 76627  0520 00 0 01713          ZET     EXCISD          DETERMINE IF COMPILER GONE
 76630  0020 00 0 76647          TRA     EXCA            YES, IT IS
 76631  0534 00 4 76651          LXA     COMPZR,4        ZERO COMPILER
 76632  0600 00 4 76626          STZ     EXCIS,4
 76633  2 00001 4 76632          TIX     *-1,4,1
 76634  0560 00 0 77056          LDQ     EXCIQ
 76635  0100 00 0 76640          TZE     EXCB            ONLY WANTS THE COMPILER
 76636 -0625 00 0 76652  EXCC     STL     EXCZR           INDICATE WANTS LAP AS WELL
 76637  0560 00 0 77057          LDQ     EXCIR
 76640 -0600 00 0 01674  EXCB     STQ     $TFS
 76641  0600 00 0 01712          STZ     BPLACE          MAKE ASSEMBLY GO TO BPS
 76642 -0625 00 0 01713          STL EXCISD               SAY COMPILER GONE
 76643 -0625 00 0 77055          STL     RST             RESETUP SWITCH
 76644  0600 00 0 15177          STZ     PROSCM          KILL COMPILER QUOTES
 76645  0600 00 0 15173          STZ     QTLSTC          AND SPECIALS
 76646  0020 00 0 76750          TRA     RSU
 76647  0100 00 4 00001  EXCA     TZE     1,4             COMPILER IS ALREADY GONE
 76650  0020 00 0 76636          TRA     EXCC            ZERO LAP AND EXCISE
 76651  0 00000 0 06366  COMPZR   PZE     EXCIS-UPERML    LENGTH OF COMPILER STORAGE
 76652  0 00000 0 00000  EXCZR    PZE                     NONZERO MEANS KILL EXCISE AND LAP
                                *
                                **CHANGES MADE TO SETUP ABOVE AND BELOW***
                                ****IT NOW ZEROES MEMORY IN ORDER TO AVOID SPURIOUS  PROTECTION OF
                                ** MEMORY FROM THE GC
                                *
 76653  0500 00 0 01703  SETUP1  CLA     $TPG            REORGANIZED SETUP******
 76654  0601 00 0 01704          STO     $ORG
 76655  0400 00 0 01705          ADD     LBINPG
 76656  0734 00 4 00000          PAX     0,4
 76657  1 77777 4 76660          TXI     *+1,4,-1
 76660 -0634 00 4 03022          SXD     C$LBPTP,4               SETUP FOR LAP
 76661  0400 00 0 15245          ADD     =1
 76662  0737 00 4 00000          PAC     0,4
 76663 -0634 00 4 01727          SXD     $CPPI,4                 SET PUSH DOWN CELLS
 76664 -0634 00 4 02002          SXD     $CSSI,4
 76665 -0634 00 4 10446          SXD     NUBPDL,4                PRIVATE COPY FOR BACKTRACE
```

```
76666   0400 00 0 01706          ADD     LPBPDL
76667   0402 00 0 15245          SUB     =1              ***PROTECTION AGAINST ZEROING FULL WORD***
76670   0621 00 0 02204          STA     ZPDL            G C ZEROS UNUSED PDL
76671   0402 00 0 15261          SUB     =19                  PROTECTION AGINST COMPILER SAVING
76672   0737 00 4 00000          PAC     0,4                  WITH OUT LOOKING
76673  -0634 00 4 02003          SXD     ENDPDL,4             OUT OF PDL TEST
76674  -0634 00 4 15157          SXD     ENPDL1,4
76675   0020 00 0 76750          TRA     RSU
76676   0634 00 4 77042  SETUP   SXA     SUPX,4          REORGANIZED SETUP******
76677   0774 00 2 15314          AXT     =0777777777777,2 GET ADDRESS OF HIGHEST LITERAL
76700   1 00001 2 76701          TXI     *+1,2,1         COMPUTE END OF PROGRAM
76701   0634 00 2 01703          SXA     $TPG,2          GIVE THIS INFORMATION TO GARBAGEMAN
76702   0774 00 2 00003          AXT     3,2
76703   0754 00 2 00000  SLOOP   PXA     0,2             ***I HAVNT SAVED IR2***
76704  -0760 00 0 00003          SSM
76705   0400 00 0 15250          ADD     =5
76706   0621 00 0 76710          STA     GETWD
76707                            TSSX    GETCOM,4        READ SECOND WORD OF COMMAND
76710   0000 00 0 00002  GETWD   HTR     2
76711   0774 00 4 00005          AXT     5,4             ASSUMES WORD NONBLANK*****
76712  -0765 00 0 00003          LGR     3
76713   0771 00 0 00003          ARS     3
76714   2 00001 4 76712          TIX     *-2,4,1         CONVERT BCD TO OCTAL,I HOPE***
76715  -0763 00 0 00017          LGL     15
76716   0621 00 2 01711          STA     LFREES+1,2      STORE IN CORRESPONDING BLOCK LENGTH
76717   2 00001 2 76703          TIX     SLOOP,2,1
76720   0500 00 0 01674          CLA     $TFS
76721   0402 00 0 01710          SUB     LFREES
76722   0621 00 0 01677          STA     $TBT
76723   0400 00 0 15245          ADD     =1
76724   0621 00 0 01676          STA     $BFS
76725   0500 00 0 01707          CLA     LFULWS
76726   0771 00 0 00005          ARS     5
76727   0400 00 0 15245          ADD     =1
76730   0601 00 0 01711          STO     $LBT
76731   0500 00 0 01676          CLA     $BFS
76732   0402 00 0 01711          SUB     $LBT
76733   0601 00 0 01700          STO     $BBT
76734   0402 00 0 15245          SUB     =1
76735   0601 00 0 01701          STO     $TFW
76736   0500 00 0 01676          CLA     $BFS
76737   0402 00 0 01707          SUB     LFULWS
76740   0601 00 0 01702          STO     $BFW
76741   0734 00 4 00000          PAX     0,4
76742   1 77777 4 76743          TXI     *+1,4,-1
76743  -0636 00 4 01730          SCD     $CPPI+1,4
                         * LETS HOPE THE ABOVE CORRECTION WORKS***
76744   0402 00 0 01706          SUB     LPBPDL
                         * SETUP WILL COMPUTE LENGTH OF BINARY PROGRAM TO MAXIMUM POSSIBLE****
76745   0402 00 0 01703          SUB     $TPG
76746   0621 00 0 01705          STA     LBINPG
76747   0020 00 0 76653          TRA     SETUP1          REORGANIZED SETUP****
                         *                STRPNT SETUP
76750                    RSU     BSS     0
                         * STRPNT HAS BEEN DELETED***********
```

```
                                *                 RECLAIMER SETUP
76750  0534 00 4 01711               LXA     $LBT,4
76751  0634 00 4 02020               SXA     A,4
76752  0534 00 4 01676               LXA     $BFS,4
76753  0634 00 4 02021               SXA     B,4
76754  0534 00 4 01701               LXA     $TFW,4
76755 -0634 00 4 02157               SXD     I,4
76756 -0634 00 4 02340               SXD     MONE,4
76757  0534 00 4 01677               LXA     $TBT,4
76760  0634 00 4 02350               SXA     MLTBT,4
76761  0534 00 4 01674               LXA     $TFS,4
76762  0634 00 4 02136               SXA     F,4
76763  0534 00 4 01676               LXA     $BFS,4
76764  0634 00 4 02171               SXA     SFWLD,4
76765  0534 00 4 01702               LXA     $BFW,4
76766  0634 00 4 02156               SXA     H,4
76767  0535 00 4 01702               LAC     $BFW,4
76770 -0634 00 4 02310               SXD     MRKLST,4
76771 -0634 00 4 02336               SXD     MLBDW,4
76772  0535 00 4 01674               LAC     $TFS,4
76773  1 77777 4 76774               TXI     *+1,4,-1
76774 -0634 00 4 02311               SXD     MRKLST+1,4
76775 -0634 00 4 02333               SXD     MLIST,4
76776  0535 00 4 01676               LAC     $BFS,4
76777 -0634 00 4 02143               SXD     G,4
77000 -0634 00 4 02334               SXD     MLBFA,4
77001 -0535 00 4 02003               LDC     ENDPDL,4
77002  1 00001 4 77003               TXI     *+1,4,1
77003  0634 00 4 02322               SXA     MLEPD,4
77004  0634 00 4 02331               SXA     MLEPE,4
77005  0535 00 4 01700               LAC     $BBT,4
77006 -0634 00 4 02335               SXD     MLBBJ,4
77007  0520 00 0 77055               ZET     RST                 SKIP IF INITAL SETUP
77010  0020 00 0 77042               TRA     SUPX                GO TO EXIT CTERWISE
77011  0535 00 4 01676               LAC     $BFS,4              BOTTOM OF FREE STORAGE
77012  1 77777 4 77013               TXI     *+1,4,-1            SUBTRACT 1
77013 -0634 00 4 77022               SXD     SUPFS,4             SET DECREMENT
77014  0535 00 4 01675               LAC     $MFS,4              LOWERP
77015 -0754 00 4 00000               PXD     0,4                 POINTER TO LWERP IN DECREMENT
77016  0601 00 0 02735               STO     $FREE               SET UP FREE
77017  0400 00 0 00126               ADD     $QD1
77020  0601 00 4 00000               STO     0,4                 START MAKING FREE STORAGE
77021  1 00001 4 77022               TXI     *+1,4,1
77022 -3 00000 4 77017         SUPFS TXL     *-3,4,**            -BFS
77023  0600 00 4 00000               STZ     0,4
77024  0535 00 4 01702               LAC     $BFW,4              BOTTOM FULL WORD SPACR
77025 -0754 00 4 00000               PXD     0,4
77026  0601 00 0 02713               STO     FWORDL              SET UP FULL WORD LIST
77027 -0737 00 4 00000               PDC     0,4                 GET IT RUE IN INDEX
77030 -0634 00 4 77033               SXD     SUPFV,4             USE TC CALCULATE LENGTH OF FULL WORD S
77031  0534 00 4 01700               LXA     $BBT,4              TFW + 1
77032  0634 00 4 77035               SXA     SUPFW,4             SET END + 1 ADDRESSS
77033  2 00000 4 77034         SUPFV TIX     *+1,4,**            LENGHT CF FULL WORD
77034  0402 00 0 00126               SUB     $QD1
77035  0601 00 4 00000         SUPFW STO     **,4                MAKE LIST
```

```
77036   2 00001 4 77034          TIX     *-2,4,1             LOOP
77037   0600 60 0 77035          STZ*    SUPFW               MAKE LAST ENTRY ZERO
77040   0500 00 0 70237          CLA     $OBLB               BEGINNING OF UNSORTED OBJECT LIST
77041   0074 00 4 77060          TSX     CNSFWL,4
77042   0774 00 4 00000   SUPX   AXT     **,4
77043   0520 00 0 76652          ZET     EXCZR           DO WE WANT TO ZERO LAP
77044   0020 00 0 77740          TRA     ZEROXC          YES, WE DO
77045   0520 00 0 77055          ZET     RST             IS IT EXCISE .
77046   0020 00 0 77051          TRA     *+3             YES IT IS
77047   0074 00 4 01145          TSX     $BPSREM,4       IT IS SETUP,WRITE NUMBER
77050   0074 00 4 03564          TSX     $PRINT,4        OF WORDS IN DECIMAL IN BPS
77051   0600 00 0 77055          STZ     RST                 ZERO RESETUP SWITCH
77052   0534 00 4 77042          LXA     SUPX,4          INDEX MAY HAVE BEEN KILLED
77053  -0754 00 0 00000          PXD     0,0
77054   0020 00 4 00001          TRA     1,4             ***OLD RETURN RESTORED***
77055   0 00000 0 00000   RST                                RESETUP TEST CELL
77056   0 00000 0 76625   EXCIQ          EXCIS-1         TOP FOR EXCISE(NIL)-REALLY MODIFIED**
77057   0 00000 0 77737   EXCIR          $BUTCH-1            TOP FOR EXCISE ( I )
```

```
                                    INSERT  $CNSFWF
                                    HEAD    E

                                *
                                * CNSFWL       USED BY SETUP TO MOVE ALL FULL WORDS ON PERMENENT OBJECTS
                                *             TO THE FULL WORD SPACE.
                                *             ALSO BUCKET SORTS THE PERMENENT OBJECTS.
                                *
77060   0634 00 4 77107  CNSFWL SXA     CNFWX,4          SAVE INDEX REGISTERS
77061   0634 00 2 77110          SXA     CNFWY,2
77062  -0734 00 4 00000          PDX     0,4              POINTER TO OBJECT LIST
77063   0500 00 4 00000   CNMLP  CLA     0,4              NEXT WORD ON LIST
77064   0622 00 0 02520          STD     CNXT             POINTER TO NEXT WORD
77065   0734 00 2 00000          PAX     0,2              POINTET TO AN ATOM
77066  -0634 00 2 02523          SXD     CNAT,2           SAVE THE POINTER TO THE ATOM
77067   0500 00 2 00000          CLA     0,2
77070  -0320 00 0 00137          ANA     TAGMSK           TEST FOR NUMBER
77071  -0100 00 0 77112          TNZ     CNNM             MAKE A NUMVER
77072   0500 00 2 00000  CNSLP  CLA     0,2              NEXT WORD ON ATOM
77073   0734 00 2 00000          PAX     0,2              CAR OF ATOM, SEARCH FOR FULL WORD
77074   3 10465 2 77076          TXH     *+2,2,$SUBR      SUCH AS $SUBR
77075   3 10464 2 77121          TXH     CMKO,2,$SUBR-1
77076   3 12065 2 77100          TXH     *+2,2,$FSUBR
77077   3 12064 2 77121          TXH     CMKO,2,$FSUBR-1
77100   3 11227 2 77102          TXH     *+2,2,$PNAME
77101   3 11226 2 77136          TXH     CMPNT,2,$PNAME-1
77102  -0734 00 2 00000  CNRS   PDX     0,2              IS NONE OF ABOVE SO CDR TO IR 2
77103   3 00000 2 77072  CNRT   TXH     CNSLP,2,0        GO BACK IF NOT END OF PROPERTY LIST
77104  -0534 00 4 02520  CNNR   LXD     CNXT,4           POINTER TO NEXT OBJECT
77105   3 00000 4 77063          TXH     CNMLP,4,0        GO BACK IF NOT END
77106  -0754 00 0 00000          PXD     0,0              CLAER AC
77107   0774 00 4 00000  CNFWX  AXT     **,4             RESTORE INDEX REGISTERS
77110   0774 00 2 00000  CNFWY  AXT     **,2
77111   0020 00 4 00001          TRA     1,4              EXIT
                                *
77112   0500 00 2 00000  CNNM   CLA     0,2
77113  -0120 00 0 77104          TMI     CNNR             DONT MOVE NUMBERS WITH MZE PREFIX
77114  -0734 00 4 00000          PDX     0,4
77115   0500 00 4 00000          CLA     0,4
77116   0074 00 4 02674          TSX     $CONSW,4
77117   0622 00 2 00000          STD     0,2
77120   0020 00 0 77104          TRA     CNNR             MAKE UP THE NEW NUMBER
                                *
77121  -0734 00 2 00000  CMKO   PDX     0,2              PUT ONE WORD IN FULL WORD SPACE
77122   0500 00 2 00000          CLA     0,2              GET NEXT WORD ON PROPERTY LIST
77123   0622 00 0 02521          STD     CNX              POINTER TO REST OF OBJECT
77124  -0120 00 0 77134          TMI     CMK              SKIP MOVING THIS WORD IF MINUS SIGN IS
77125   0734 00 4 00000          PAX     0,4              SENSED, OTHERWISE GET POINTER TO FULL
77126   0500 00 4 00000          CLA     0,4              WORD AND WORD IT SELF IN AC
77127   0074 00 4 02674          TSX     $CONSW,4         PUT IT IN FULL WORD SPACE
77130   0771 00 0 00022          ARS     18               MOVE POINTER TO WORD IN FWS TO ADDRESS
77131   0621 00 2 00000          STA     0,2              REPLACE THE ADDRESS
77132  -0534 00 2 02521          LXD     CNX,2            POINTER TO NEXT WORD ON PROPERTY LIST
77133   0020 00 0 77103          TRA     CNRT             RETURN
                                *
```

```
77134  0602 00 2 00000   CMK    SLW     0,2          RESTORE WORD WITH PLUS SIGN
77135  0020 00 0 77102          TRA     CNRS         GO BACK
                *
77136 -0734 00 2 00000   CMPNT  PDX     0,2          PUT PRINT NAME IN FULL WORD SPACE
77137  0500 00 2 00000          CLA     0,2          NEXT WORD ON PROPERTY LIST
77140  0622 00 0 02521          STD     CNX          POINTER TO NEXT WORD ON PROPERTY LIST
77141  0734 00 2 00000          PAX     0,2          POINTET TO PNAME LIST
77142 -0634 00 2 02524          SXD     CNVA,2       SAVE IT
77143  0500 00 2 00000   CMPLP  CLA     0,2          FIRST FORD ON PNAME LIST
77144 -0120 00 0 77155          TMI     CMPS         SKIP IF WORD IS FLAGGED
77145  0622 00 0 02522          STD     CNFT         POINTER TO NEXT WORD ON PNAME LIST
77146  0734 00 4 00000          PAX     0,4          POINTER TO FULL WORD
77147  0500 00 4 00000          CLA     0,4          FULL WORD
77150  0074 00 4 02674          TSX     $CONSW,4     PUT IN FULL WORD SPACE
77151  0771 00 0 00022          ARS     18           POINTER TO WORD
77152  0621 00 2 00000          STA     0,2          RPLACE THE ADDRESS
77153 -0534 00 2 02522          LXD     CNFT,2       POINTER TO NEXT WORD ON PNAME LIST
77154  3 00000 2 77143          TXH     CMPLP,2,0    GO BACK IF NOT END
77155  0500 00 0 02524   CMPS   CLA     CNVA         POINTER TO PNAME LIST
77156  0560 00 0 02523          LDQ     CNAT         ATOM THAT WE ARE WORKING ON
77157  0074 00 4 06405          TSX     BUKSRT,4     PUT ON BUCKET SORTED OBJECT LIST
77160 -0534 00 2 02521          LXD     CNX,2        POINTER TO NEXT WORD ON ATOM
77161  0020 00 0 77103          TRA     CNRT         GO BACK
                         INSERT  $LAPZFE
```

```
                * ORG DELETED WILL RESIDE AFTER SETUP
                      HEAD    C                    THIS IS THE COMPILER AND ASMBLR
                *
                * LAP IS THE ASSEMBLER. ONE ARG IS LISTING.  IT IS LIST OF INSTRUC-
                * TIONS, NON-ATOMIC OR NIL. THE ATOMIC SYMBOLS ARE LOCATION SYMBOLS
                * SECOND ARG IS START OF SYMBIL TABLE WHICH IS AN A-LIST.
                * THE FIRST ITEM IS ORG AS FOLLOWS-
                *       NIL= IN BPS
                *       ATOM= AT SYMBOLIC LOCATION
                *       NUM= ATTHIS NUMBER
                *       (NAME TYPE NUM) = IN BPS, AND PUT TXL ON PROP LIST OF NAME
                *       WITH FLAG TYPE AND NUM IB DEC. OF TXL.
                * INSTRUCTION FORMAT IS (OP ADDR TAG DEC)
                * FIELD FORMAT IS AS FOLLOWS-
                *       TEMP SYMBOL
                *       NUMBER
                *       SYM SUBR OR FSUBR
                *       (E NAME) FOR IMMEDIATE AS IN TXL FILTER
                *       (QUOTE NAME) FOR ITEM IN DEC OF WORD ON QTLST
                *       POINTER TO COMMON WORD.MAKES ONE IF NONE ALREADY
                *       SUM OF ANY OF ABOVE
                * LAP IS IDENTITY FUNCTION
                *       LAP DOES NOT USE IX1. IX2,4 ARE SCARTCH
                * ERRORS IN LAP AS FOLLOWS-
                *       *L   1*  UNABLE TO EVALUATE ORIGIN
                *       *L   2*  OUT OF BPS DISCOVERED AFTER PASS I
                *       *L   3* UNDEFINED SYMBOL
                *       *L   4*  FIELD WAS RECURSIVE
                *
77162   0634 00 4 77342  LAP     SXA     LAX,4
77163   0634 00 2 77343          SXA     LAX+1,2
77164   0601 00 0 15171          STO     LIST            THIS IS THE INPUT
77165  -0600 00 0 15174          STQ     TAB             START OF SYMBOL TABLE
77166  -0734 00 4 00000          PDX     0,4
77167   0500 00 4 00000          CLA     0,4
77170   0622 00 0 77723          STD     REST            SAVE REST OF LISTING
77171   0734 00 2 00000          PAX     0,2             ORIGIN IN IX2
77172  -3 00000 2 77217          TXL     INBP,2,0        NIL MEANS  BPS ASSEMBLY
77173   0500 00 2 00000          CLA     0,2
77174   0734 00 4 00000          PAX     0,4             CAR OF ORIGIN
77175  -3 77776 4 77217          TXL     INBP,4,-2       NOT ATOM MEANS BPS MODE SO GO
77176  -0625 00 0 77727          STL     MODE            NOISE = NOT BPS
77177  -0754 00 2 00000          PXD     0,2             MAKE NUMBER TEST
77200   0074 00 4 12672          TSX     NUMBRP,4
77201  -0100 00 0 77212          TNZ     LSQ             IF A NUMBER
77202  -0754 00 2 00000          PXD     0,2             ORIGIN TO AC
77203   0560 00 4 00155          LDQ     $QSYMD,4        (QUOTE SYM)
77204   0074 00 4 10547          TSX     GET,4
77205  -0100 00 0 77212          TNZ     LSQ             ORIGIN WAS FOUND
77206  -0754 00 2 00000          PXD     0,2             SHOW IT
77207                    ERROR   LAPORG          UNDEFINED ORIGIN
77212  -0754 00 2 00000  LSQ     PXD     0,2
77213   0074 00 4 12567          TSX     NUMVAL,4        GET NUMERICAL VALUE
77214  -0734 00 4 00000          PDX     0,4
77215   0500 00 4 00000          CLA     0,4             PUTS SYM IN AC FOR NOT BPS MODE
```

```
77216  0020 00 0 77225          TRA     LAPP1
77217 -0520 00 0 01712  INBP     NZT     BPLACE          TEST IF EXCISABLE PROGRAM
77220  0500 00 0 01704          CLA     $ORG
77221  0520 00 0 01712          ZET     BPLACE
77222  0500 00 0 01673          CLA     $XORG
77223  0600 00 0 77727          STZ     MODE            INDICATES BPS MODE
77224  0074 00 4 77351          TSX     JUST,4          JUSTIFY AC
77225  0601 00 0 77724  LAPP1   STO     STAR            UPDATE MARKER
77226  0601 00 0 77725          STO     START           RESET MARKER
77227  0600 00 0 77726          STZ     PASWD           INDICATE PASS 1
77230  0074 00 4 77357          TSX     PASS,4
77231  0500 00 0 15174          CLA     TAB
77232  0520 00 0 00077          ZET     COMPWD          DO WE WANT COMPILER PRINTOUT
77233  0074 00 4 03564          TSX     $PRINT,4        YES
77234  0520 00 0 77727          ZET     MODE
77235  0020 00 0 77243          TRA     LAPP2
77236  0534 00 4 77724          LXA     STAR,4
77237 -0520 00 0 01712          NZT     BPLACE          TEST FOR OUT OF BPS
77240  0522 00 0 03022          XEC     LBPTP
77241  0520 00 0 01712          ZET     BPLACE
77242  0522 00 0 03021          XEC     LTBPFJ          CHECK
77243  0534 00 4 77725  LAPP2   LXA     START,4         RESET STAR FOR SECOND PASS
77244  0634 00 4 77724          SXA     STAR,4
77245 -0534 00 4 15171          LXD     LIST,4
77246  0500 00 4 00000          CLA     0,4
77247  0622 00 0 77723          STD     REST            USED BY PASS AGAIN
77250 -0625 00 0 77726          STL     PASWD           NOISE MEANS PASS 2
77251  0074 00 4 77357          TSX     PASS,4          FOR PASS 2
77252  0520 00 0 77727          ZET     MODE
77253  0020 00 0 77336          TRA     LEND            IF NOT IN BPS MODE
77254  0534 00 4 77724          LXA     STAR,4          RSET ORG FOR NEXT ASSEMBLY
77255 -0520 00 0 01712          NZT     BPLACE          TEST IF  BPS OR EXCISABLE AREA
77256  0634 00 4 01704          SXA     $ORG,4
77257  0520 00 0 01712          ZET     BPLACE
77260  0634 00 4 01673          SXA     $XORG,4
77261 -0534 00 4 15171          LXD     LIST,4
77262  0500 00 4 00000          CLA     0,4             CWR OF LISTING
77263  0734 00 4 00000          PAX     0,4
77264  0500 00 4 00000          CLA     0,4             GETS CWR OF ORIGIN
77265  0734 00 2 00000          PAX     0,2             CAR OF ORIGIN
77266 -0734 00 4 00000          PDX     0,4             CDR OF ORIGIN
77267  3 77776 2 77336          TXH     LEND,2,-2       IF ATOM THEN NO TXL NEEDED
77270 -0634 00 2 77345          SXD     NAME,2          CAR OF ORG IS NAME
77271  0500 00 4 00000          CLA     0,4
77272  0734 00 2 00000          PAX     0,2             CADR OF ORIGIN IS TYPE
77273 -0634 00 2 77346          SXD     TYPE,2          STORE TYPE
77274 -0320 00 0 00133          ANA     $DMASK          CDDR IS NOE IN AC
77275  0074 00 4 10470          TSX     CADARX,4        CADAR PUTS PART OF NUM IN DECR OF AC
77276  0622 00 0 77347          STD     INDC            FOR TXL WORD
77277  0534 00 4 77725          LXA     START,4
77300  0634 00 4 77347          SXA     INDC,4          COMPLETES TXL WORD
77301  0500 00 0 77345          CLA     NAME
77302  0560 00 0 77346          LDQ     TYPE
77303  0074 00 4 10547          TSX     GET,4
77304  0100 00 0 77316          TZE     MKIND           IF THERE WAS NO OLD TXL
```

```
77305 -0734 00 2 00000          PDX     0,2             SAVE POINTER TO TXL
77306  0500 00 2 00000          CLA     0,2             CWR OF CLD TXL
77307  0737 00 4 00000          PAC     0,4             POINTER TO OLD BIN PTROG.
77310  0500 00 0 77725          CLA     START           START OF NEW PROGRAM
77311 -0501 00 0 77350          ORA     PATCH           MAKE TRA INSTRUCTION
77312  0601 00 4 00000          STO     0,4             CLOBBER OLD PROG.
77313  0500 00 0 77347          CLA     INDC
77314  0601 00 2 00000          STO     0,2             ON TOP OF OLD TXL
77315  0020 00 0 77336          TRA     LEND
77316  0500 00 0 77347  MKIND   CLA     INDC
77317 -0534 00 4 77346          LXD     TYPE,4          SYM SHOULD HAVE NO TXL ON POINTER
77320 -3 10445 4 77323          TXL     IND2,4,$SYM-1
77321  3 10446 4 77323          TXH     IND2,4,$SYM
77322 -0320 00 0 00132          ANA     $AMASK
77323  0074 00 4 02674  IND2    TSX     $CONSW,4
77324  0131 00 0 00000          XCA                     SAVE AC
77325 -0534 00 2 77345          LXD     NAME,2          NAME OF SUBR OR TYPE
77326  0500 00 2 00000          CLA     0,2
77327 -0320 00 0 00133          ANA     $DMASK          CDR OF NAME NOW IN AC
77330  0131 00 0 00000          XCA
77331  0074 00 4 02714          TSX     $CONS,4         CONS (TXL,RESTOF PROPERTY LIST)
77332  0131 00 0 00000          XCA
77333  0500 00 0 77346          CLA     TYPE
77334  0074 00 4 02714          TSX     $CONS,4         CONS,TYPE,RST OF ATM)
77335  0622 00 2 00000          STD     0,2             RPLACD OF PROPERTY LIST
77336  0500 00 0 15174  LEND    CLA     TAB
77337  0600 00 0 15171          STZ     LIST
77340  0600 00 0 15174          STZ     TAB
77341  0600 00 0 77722          STZ     INST
                        *       DONT STORE ZERO IN QTLST
77342  0774 00 4 00000  LAX     AXT     **,4
77343  0774 00 2 00000          AXT     **,2
77344  0020 00 4 00001          TRA     1,4
                        * ALL LAP REGISTERS FOLLOW,INCL. THOSE USED BY SUBROUTINES
77345  0 00000 0 00000  NAME                            NAME OF FUNCTION
77346  0 00000 0 00000  TYPE                            SUBR FSUBR ETC
77347 -3 00000 0 00000  INDC    TXL     **,,**          FOR TSL WORD
77350  0020 00 0 00000  PATCH   TRA     **              FOR CLOBBER INSTRUCTION
                        *       ADDR(REM)=IX4 SAVED.DECR=REST OF LIST FIELD
                        *
                        *  JUST REDUCES THE AC MOD 2**15.THE RESULT IS 15 BITS IN ADDR OF AC
                        *    IT IS ALWAYS POSITIVE
77351  0120 00 0 77354  JUST    TPL     **+3
77352  0760 00 0 00006          COM
77353  0402 00 0 15245          SUB     =1
77354 -0320 00 0 00132          ANA     $AMASK
77355 -0140 00 4 00001          TNO     1,4
77356  0020 00 4 00001          TRA     1,4
                        *
                        *  PASS DOES BOTH PASSES FOR LAP
                        *       FIRST PASS MAKES SYMBOL TABLE AND UPDATES ON INSTRUCTIONS
                        *       SECOND PASS IGNORES SYMBOLS ASSEMBLES AND UPDTS INSTRUCTIONS
77357  0634 00 4 77414  PASS    SXA     PAUX,4
77360 -0534 00 4 77723  LOP1    LXD     REST,4
77361 -3 00000 4 77414          TXL     PAUX,4,0        IF NO MORE LISTING
```

```
77362  0500 00 4 00000          CLA     0,4
77363  0622 00 0 77723          STD     REST            RESET REST OF LISTING
77364  0734 00 4 00000          PAX     0,4
77365 -0634 00 4 77722          SXD     INST,4
77366 -3 00000 4 77406          TXL     AMBL,4,0        IF NIL
77367  0500 00 4 00000          CLA     0,4
77370  0734 00 4 00000          PAX     0,4
77371 -3 77776 4 77406          TXL     AMBL,4,-2       IF NOT ATOM
77372  0520 00 0 77726          ZET     PASWD
77373  0020 00 0 77360          TRA     LOP1            IF PASS 2
77374  0500 00 0 77724          CLA     STAR            OTHERWISE ADD    TO TABLE
77375  0560 00 0 00150          LDQ     $OCTD
77376  0074 00 4 11163          TSX     $MKNO,4         MAKE A NUMBER
77377  0131 00 0 00000          XCA
77400  0500 00 0 77722          CLA     INST
77401  0074 00 4 02714          TSX     $CONS,4         (NAME.VALUE)
77402  0560 00 0 15174          LDQ     TAB
77403  0074 00 4 02714          TSX     $CONS,4
77404  0601 00 0 15174          STO     TAB
77405  0020 00 0 77360          TRA     LOP1
77406  0520 00 0 77726  AMBL    ZET     PASWD           LAND HERE IF INSTRUCTION NOT SYMBOL
77407  0074 00 4 77416          TSX     AINS,4          ON PASS 2 ONLY
77410  0534 00 4 77724          LXA     STAR,4
77411  1 00001 4 77412          TXI     *+1,4,1         UPDATE * AFTER INSTRUCTION IS ASSEMBLE
77412  0634 00 4 77724          SXA     STAR,4
77413  0020 00 0 77360          TRA     LOP1
77414  0774 00 4 00000  PAUX    AXT     **,4
77415  0020 00 4 00001          TRA     1,4
                                *
                                * AINS IS THE INSTRUCTION ASSEMBLER. ARG IS IN INST. VAL IS IN AC
77416  0634 00 4 77434  AINS    SXA     AINX,4
77417  0074 00 4 77436          TSX     AFELD,4
77420  0601 60 0 77724          STO*    STAR
77421  0074 00 4 77436          TSX     AFELD,4
77422  0074 00 4 77351          TSX     JUST,4
77423 -0602 60 0 77724          ORS*    STAR            THIS IS ADDRESS FIELD
77424  0074 00 4 77436          TSX     AFELD,4
77425  0767 00 0 00017          ALS     15
77426  0140 00 0 77427          TOV     *+1
77427 -0602 60 0 77724          ORS*    STAR            TAG FIELD
77430  0074 00 4 77436          TSX     AFELD,4
77431  0074 00 4 77351          TSX     JUST,4
77432  0767 00 0 00022          ALS     18              NO OVERFLOW AFTER JUST
77433 -0602 60 0 77724          ORS*    STAR
77434  0774 00 4 00000  AINX    AXT     **,4
77435  0020 00 4 00001          TRA     1,4
                                *
                                * AFELD IS THE FIELD EVALUATOR. A LIST OF FIELDS IS EXPECTED IN INST.
                                * IT EVALUATES THE FIRST AND SETS INST TO THE REST. IF NO MORE FIELDS LE
                                * ARE LEFT, IT GOES TOAINX, THE EXIT POINT OF AINS
                                * AFELD HAS CERTAIN PRIVATE CELLS,SEE AFTER LAP.) THE LIST AFELD IS A
                                * SLIGHTLY RECURSIVE DEVISE WHICH HAS SPECIAL CELLS AND CANNOT REENTER I
                                * ITSELF WITHOUT ERROR.
77436  0634 00 4 77663  AFELD   SXA     FELX,4
77437  0600 00 0 77732          STZ     NOCUR
```

```
77440 -0534 00 4 77722          LXD     INST,4
77441 -3 00000 4 77434          TXL     AINX,4,0        IF NO MORE FIELDS
77442  0500 00 4 00000          CLA     0,4
77443  0622 00 0 77722          STD     INST            REST OF FIELDS
77444  0734 00 2 00000          PAX     0,2
77445  0500 00 2 00000  LEM     CLA     0,2
77446  0734 00 4 00000          PAX     0,4
77447 -3 77776 4 77526          TXL     NATM,4,-2       IF NOT ATOMIC FIELD
77450  3 00000 2 77456          TXH     LAPP3,2,0
77451 -0520 00 0 01712          NZT     BPLACE          TEST IF  BPS OR EXCISABLE AREA
77452  0500 00 0 01704          CLA     $ORG
77453  0520 00 0 01712          ZET     BPLACE          TEST IF  BPS OR EXCISABLE AREA
77454  0500 00 0 01673          CLA     $XORG
77455  0020 00 0 77663          TRA     FELX
77456 -0754 00 2 00000  LAPP3   PXD     0,2
77457  0560 00 0 77735          LDQ     LSAC            FN ARG FOR SASSOC
77460 -0600 00 0 02531          STQ     $ARG3
77461  0560 00 0 15174          LDQ     TAB
77462  0074 00 4 07663          TSX     SASSOC,4        LOOK UP IN SYM TABLE
77463  0100 00 0 77471          TZE     NTAB            NOT IN TAB
77464  0074 00 4 13451          TSX     $CORP,4
77465  0074 00 4 12567  NEVAL   TSX     NUMVAL,4
77466 -0734 00 4 00000          PDX     0,4
77467  0500 00 4 00000          CLA     0,4
77470  0020 00 0 77663          TRA     FELX
77471 -0754 00 2 00000  NTAB    PXD     0,2
77472  0074 00 4 12672          TSX     NUMBRP,4
77473  0100 00 0 77476          TZE     *+3                     IF NOT A NUMBER
77474 -0754 00 2 00000          PXD     0,2                     LISP NUMBER IN AC
77475  0020 00 0 77465          TRA     NEVAL
77476 -0634 00 2 77734          SXD     ERCC,2                  SAVE ATOM
77477  0500 00 2 00000  LOP2    CLA     0,2     LOOP FOR SYM,SUBR,FSUBR
77500 -0734 00 2 00000          PDX     0,2
77501  0734 00 4 00000          PAX     0,4
77502  3 00000 2 77507          TXH     PA,2,0                  IF NOT NIL
77503  0500 00 0 77734          CLA     ERCC
77504                            ERROR   LAPSMB          UNDEFINED SYMBOL
77507 -3 10445 4 77511  PA      TXL     *+2,4,$SYM-1
77510 -3 10446 4 77522          TXL     FINX,4,$SYM
77511 -3 10464 4 77513          TXL     *+2,4,$SUBR-1
77512 -3 10465 4 77515          TXL     FIND,4,$SUBR
77513 -3 12064 4 77477          TXL     LOP2,4,$FSUBR-1
77514  3 12065 4 77477          TXH     LOP2,4,$FSUBR
77515  0500 00 2 00000  FIND    CLA     0,2
77516  0734 00 4 00000          PAX     0,4
77517  0500 00 4 00000          CLA     0,4
77520 -0320 00 0 00132          ANA     $AMASK
77521  0020 00 0 77663          TRA     FELX
                        *
77522  0500 00 2 00000  FINX    CLA     0,2
77523  0734 00 4 00000          PAX     0,4
77524  0500 00 4 00000          CLA     0,4
77525  0020 00 0 77663          TRA     FELX
                        *
77526 -3 07666 4 77535  NATM    TXL     NTE,4,$H25-1
```

```
77527  3 07667 4 77535        TXH     NTE,4,$H25          FOR (E EXP)
77530 -0754 00 2 00000        PXD     0,2                 ENTIRE FIELD
77531  0074 00 4 10506        TSX     CADRXX,4
77532  0074 00 4 77665        TSX     PRO,4                   PROTECT LITERAL ON PROG LIST
77533  0771 00 0 00022        ARS     18
77534  0020 00 0 77663        TRA     FELX
77535 -3 11077 4 77604 NTE    TXL     NQT,4,$QUOTE-1      LAND HERE FOR NOT (E...
77536  3 11100 4 77604        TXH     NQT,4,$QUOTE        ABOVE AND THIS FOR (QUOTE...
77537 -0734 00 2 00000        PDX     0,2                 AC HAS CWR OF FIELD
77540  0500 00 2 00000        CLA     0,2
77541  0734 00 2 00000        PAX     0,2                 POINTER TO EQ QUANTITY
77542 -0534 00 4 15172        LXD     QTLST,4
77543  0520 00 0 01712        ZET     BPLACE
77544 -0534 00 4 15173        LXD     QTLSTC,4        WHEN COMPILER IS IN
77545 -3 00000 4 77561        TXL     NON,4,0             TEST FOR NO LIST
77546  0500 00 4 00000 FLOOP  CLA     0,4                 AN EQUAL TYPE SEARCH
77547  0601 00 0 77730        STO     HOLD                TEMPORARY SAVING OF REST
77550  0734 00 4 00000        PAX     0,4
77551  0500 00 4 00000        CLA     0,4
77552 -0320 00 0 00133        ANA     $DMASK              LITERAL QUANTITY FOR EQUAL COMPARISON
77553  0131 00 0 00000        XCA
77554 -0754 00 2 00000        PXD     0,2                 THE NEW ITEM
77555  0074 00 4 03450        TSX     $EQUAL,4            TEST FOR EQUALITY
77556 -0100 00 0 77602        TNZ     ONQT                IF ALREADY ON LIST
77557 -0534 00 4 77730        LXD     HOLD,4
77560  3 00000 4 77546        TXH     FLOOP,4,0           IF NOT HEAD OF QTLIST
77561 -0754 00 2 00000 NON    PXD     0,2                 NEED TO MAKE ENTRY
77562  0131 00 0 00000        XCA
77563  0754 00 0 00000        PXA     0,0
77564  0074 00 4 02714        TSX     $CONS,4             CONS(NIL EXP)
77565  0601 00 0 77730        STO     HOLD                NEEDS NO PROTECTION AS SEEN BY WHAT FO
                       *              FOLLOWS
77566  0560 00 0 15172        LDQ     QTLST
77567  0520 00 0 01712        ZET     BPLACE
77570  0560 00 0 15173        LDQ     QTLSTC          WHEN COMPILER IS READ IN****
77571  0074 00 4 02714        TSX     $CONS,4             CONS((NIL.EXP, ...
77572  0520 00 0 01712        ZET     BPLACE
77573  0020 00 0 77576        TRA     *+3
77574  0601 00 0 15172        STO     QTLST           BINARY PROG
77575  0020 00 0 77577        TRA     *+2
77576  0601 00 0 15173        STO     QTLSTC          COMPILER SPACE
77577 -0535 00 4 77730        LDC     HOLD,4              WANT TRUE POINTER
77600  0754 00 4 00000 TRP    PXA     0,4
77601  0020 00 0 77663        TRA     FELX
                       *      THIS IS POINTER TO A NIL.EXP WORD IN FREE STORAGE
77602  0535 00 4 77730 ONQT   LAC     HOLD,4
77603  0020 00 0 77600        TRA     TRP
77604 -3 10537 4 77635 NQT    TXL     FDLST,4,SPECAL-1
77605  3 10540 4 77635        TXH     FDLST,4,SPECAL      (SPECIAL NAME)
77606  0560 00 0 00154        LDQ     QSPECD              SPECIAL IN MQ
77607 -0320 00 0 00133        ANA     $DMASK              (NAME) IN AC
77610  0074 00 4 13441        TSX     $CARP,4
77611  0074 00 4 10547        TSX     GET,4
77612 -0100 00 0 77632        TNZ     SPP                 JUST NEED TO ASSURE PROTECTION
77613  0560 00 0 15244        LDQ     =0
```

```
77614  0074 00 4 02714          TSX     $CONS,4          AC HAS ZERO IF YOU ARE HERE
77615  0601 00 0 15175          STO     LCOM             PROTECTED TEMP CELL
                        * BY SOME GHASTLY MISTAKE MY FILE WAS SHORTENED
                        * I AM NOW MAKING A VALIANT EFFORT TO ELONGATE IT.
77616 -0754 00 2 00000          PXD     0,2              (SPECIAL NAME)
77617  0074 00 4 10506          TSX     CADRXX,4
77620 -0734 00 2 00000          PDX     0,2
77621  0074 00 4 13451          TSX     $CDRP,4
77622  0131 00 0 00000          XCA
77623  0500 00 0 15175          CLA     LCOM
77624  0074 00 4 02714          TSX     $CONS,4
77625  0131 00 0 00000          XCA
77626  0500 00 0 00154          CLA     QSPECD
77627  0074 00 4 02714          TSX     $CONS,4
77630  0622 00 2 00000          STD     0,2              RPLACD OF NAME
77631  0074 00 4 10506          TSX     CADRXX,4         POINTER TO (NIL)
77632  0074 00 4 77665  SPP     TSX     PRO,4
77633  0737 00 4 00000          PDC     0,4
77634  0020 00 0 77600          TRA     TRP
77635  0520 00 0 77732  FDLST   NZT     NOCUR            NO REENTRY TO AFELD LIST IS ALLOWED
77636  0020 00 0 77642          TRA     *+4
77637                    ERROR   LAPSUB          NO RECURSIVE SUBFIELDS ALLOWED
77642 -0625 00 0 77732          STL     NOCUR            PREVENT RECURSION
77643  0600 00 0 77731          STZ     SUM              REST SUM WORD
77644  0534 00 4 77663          LXA     FELX,4
77645  0634 00 4 77733          SXA     REM,4            SAVES THE RETURN FOR AFELD
77646  0500 00 2 00000  LOPL    CLA     0,2
77647  0622 00 0 77733          STD     REM
77650  0734 00 2 00000          PAX     0,2
77651 -0774 00 4 77653          AXC     *+2,4
77652  0634 00 4 77663          SXA     FELX,4           REENTER THE EVALUATOR
77653  0020 00 0 77445          TRA     LEM
77654  0400 00 0 77731          ADD     SUM
77655  0601 00 0 77731          STO     SUM
77656 -0534 00 2 77733          LXD     REM,2            REST OF FIELDS
77657  3 00000 2 77646          TXH     LOPL,2,0         IF THERE ARE MORE FIELDS(SUBFIELDS)
77660  0600 00 0 77732          STZ     NOCUR            ALLOWS ENTRY TO LIST AFELD AGAIN
77661  0534 00 4 77733          LXA     REM,4
77662  0020 00 4 00001          TRA     1,4
77663  0774 00 4 00000  FELX    AXT     **,4
77664  0020 00 4 00001          TRA     1,4
                        *
77665  0634 00 4 77703  PRO     SXA     PX,4
77666  0601 00 0 77721          STO PTR SAVE ARGUMENT
77667  0622 00 0 77702          STD     PH               SET UP TXH
77670  0402 00 0 00126          SUB     $QD1             AND
77671  0622 00 0 77701          STD     PL               TXL  CLA PROS GET PROTECTED LIST
77672  0500 00 0 15176          CLA     PROS             ***DELETED BY MY ERROR***, THE PROPHET
77673  0520 00 0 01712          ZET     BPLACE
77674  0500 00 0 15177          CLA     PROSCM           WHEN COMPILER IS BEING READ IN***
77675 -0734 00 4 00000  PNL     PDX     0,4
77676 -3 00000 4 77706          TXL     PMK,4,0          END OF LIST,SO MAKE NEW ENTRY
77677  0500 00 4 00000          CLA     0,4
77700  0734 00 4 00000          PAX     0,4
77701 -3 00000 4 77675  PL      TXL     PNL,4,**
```

```
77702  3 00000 4 77675  PH      TXH   PNL,4,**     FALL THROUGH IF FOUND
77703  0774 00 4 00000  PX      AXT     **,4
77704  0500 00 0 77721          CLA     PTR          RESTORE AC
77705  0020 00 4 00001          TRA     1,4
                         *
77706  0560 00 0 15176  PMK     LDQ     PROS         MAKE A NEW ITEM
77707  0520 00 0 01712          ZET     BPLACE
77710  0560 00 0 15177          LDQ     PROSCM       WHEN COMPILER IS BEING READ IN***
77711  0500 00 0 77721          CLA     PTR
77712  0074 00 4 02714          TSX     $CONS,4
77713  0520 00 0 01712          ZET     BPLACE
77714  0020 00 0 77717          TRA     *+3
77715  0601 00 0 15176          STO     PROS         BINARY PROGRAM SPACE LIST
77716  0020 00 0 77703          TRA     PX           RETURN
77717  0601 00 0 15177          STO     PROSCM       .OMPILER LIST
77720  0020 00 0 77703          TRA     PX           RETURN
                         *
77721  0 00000 0 00000  PTR
                         *
                         *
77722  0 00000 0 00000  INST                         HOLDS CURRENT INSTRUCTION OR FRACTION
77723  0 00000 0 00000  REST                         REMAINDER OF LISTING.PASS ALTERS THIS
77724  0 00000 0 00000  STAR                         * DIRECT ADDRESS POINTER TO CURRENT LO
77725  0 00000 0 00000  STARI                        RESET CELL FOR *
77726  0 00000 0 00000  PASWD                        ZERO MEANS PASS 1.NOISE MEANS PASS 2
77727  0 00000 0 00000  MODE                         ZERO MEANS BPS ASSEMBLY
77730  0 00000 0 00000  HOLD                         SCRATCH CALL FOR AFELD ONLY,WATCH OUT
77731  0 00000 0 00000  SUM                          FOR USE BY AFELD LIST ONLY
77732  0 00000 0 00000  NOCUR                        FOR AFELD LIST ONLY,PREVENTS RECURSION
77733  0 00000 0 00000  REM                          FOR AFELD LIST ONLY
77734  0 00000 0 00000  ERCC
77735 -3 00000 0 77736  LSAC    TXL     *+1,,0
T      77736 -0754 00 0 00000           PXD
       77737  0020 00 4 00001          TRA     1,4
                                       HEAD    0
```

```
                                *
                                *  BUTCH , A HOME FOR PATCHES
                                *
77740                           BUTCH   BSS     0               ***CHANGE HERE***
77740  0634 00 4 77744          ZEROXC  SXA     ZEROX,4         SAVE IR 4
77741  0774 00 4 01112                  AXT     BUTCH-E$EXCIS,4
77742  0600 00 4 77740                  STZ     BUTCH,4         ZEROS LAP AND EXCISE
77743  2 00001 4 77742                  TIX     *-1,4,1
77744  0774 00 4 00000          ZEROX   AXT     **,4
77745 -0754 00 0 00000                  ZAC                     PREPARE RESULT OF NIL
77746  0020 00 4 00001                  TRA     1,4
```

```
77747                                  BSS     LITRLS-*    GET READY TO DUMP THE LITERALS AND RMTS
                                       HEAD    0
15204                                  RMT     *           HERE COME THE REMOTES
15204                                  TSSX1   SETMEM                                  .C01
15204  0101 00 0 15205  SETMEM         TIA     *+1                                     .C02
15205  622563442544                    BCI     1,SETMEM                                .C02
15206                                  TSSX1   WRFLX                                   .C01
15206  0101 00 0 15207  WRFLX          TIA     *+1                                     .C02
15207  665126436760                    BCI     1,WRFLX                                 .C02
15210                                  TSSX1   DORMNT                                  .C01
15210  0101 00 0 15211  DORMNT         TIA     *+1                                     .C02
15211  244651444563                    BCI     1,DORMNT                                .C02
15212                                  TSSX1   GETCOM                                  .C01
15212  0101 00 0 15213  GETCOM         TIA     *+1                                     .C02
15213  272563234644                    BCI     1,GETCOM                                .C02
15214                                  TSSX1   DELFIL                                  .C01
15214  0101 00 0 15215  DELFIL         TIA     *+1                                     .C02
15215  242543263143                    BCI     1,DELFIL                                .C02
15216                                  TSSX1   OPEN                                    .C01
15216  0101 00 0 15217  OPEN           TIA     *+1                                     .C02
15217  464725456060                    BCI     1,OPEN                                  .C02
15220                                  TSSX1   BUFFER                                  .C01
15220  0101 00 0 15221  BUFFER         TIA     *+1                                     .C02
15221  226426262551                    BCI     1,BUFFER                                .C02
15222                                  TSSX1   CLOSE                                   .C01
15222  0101 00 0 15223  CLOSE          TIA     *+1                                     .C02
15223  234346622560                    BCI     1,CLOSE                                 .C02
15224                                  TSSX1   RDFLXA                                  .C01
15224  0101 00 0 15225  RDFLXA         TIA     *+1                                     .C02
15225  512426436721                    BCI     1,RDFLXA                                .C02
15226                                  TSSX1   FSTATE                                  .C01
15226  0101 00 0 15227  FSTATE         TIA     *+1                                     .C02
15227  266263216325                    BCI     1,FSTATE                                .C02
15230                                  TSSX1   SAVBRK                                  .C01
15230  0101 00 0 15231  SAVBRK         TIA     *+1                                     .C02
15231  622165225142                    BCI     1,SAVBRK                                .C02
15232                                  TSSX1   SETBRK                                  .C01
15232  0101 00 0 15233  SETBRK         TIA     *+1                                     .C02
15233  622563225142                    BCI     1,SETBRK                                .C02
15234                                  TSSX1   GETBRK                                  .C01
15234  0101 00 0 15235  GETBRK         TIA     *+1                                     .C02
15235  272563225142                    BCI     1,GETBRK                                .C02
15236                                  TSSX1   WRWAIT                                  .C01
15236  0101 00 0 15237  WRWAIT         TIA     *+1                                     .C02
15237  665166213163                    BCI     1,WRWAIT                                .C02
15240                                  TSSX1   RDWAIT                                  .C01
15240  0101 00 0 15241  RDWAIT         TIA     *+1                                     .C02
15241  512466213163                    BCI     1,RDWAIT                                .C02
15242                                  TSSX1   CHNCOM                                  .C01
15242  0101 00 0 15243  CHNCOM         TIA     *+1                                     .C02
15243  233045234644                    BCI     1,CHNCOM                                .C02
```

00024 END 20 20 IS WHERE IT ALL BEGINS

LITERALS

15244 000000000000
15245 000000000001
15246 000000000002
15247 000000000004
15250 000000000005
15251 000000000006
15252 000000000007
15253 000000000010
15254 000000000012
15255 000000000014
15256 000000000016
15257 000000000020
15260 000000000022
15261 000000000023
15262 000000000025
15263 000000000033
15264 000000000040
15265 000000000050
15266 000000000053
15267 000000000057
15270 000000000060
15271 000000000077
15272 000000000100
15273 000000000200
15274 000000000660
15275 201400000000
15276 214343606060
15277 270000000000
15300 273165256447
15301 430000000000
15302 454623464463
15303 516060606060
15304 600000000000
15305 603434343434
15306 605325462653
15307 606043316247
15310 606060605353
15311 606060606060
15312 606521436425
15313 666060606060
15314 777777777777

POST PROCESSOR ASSEMBLY DATA

15315 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

REFERENCES TO DEFINED SYMBOLS

5	N	64466,64477,64511,64522,64533,64544,64552,64563,64574,64605,64616,64631,64642,64655,64666,64677 64721,64732,64743,64754,64765,64776,65007,65020,65031,65042,65054,65067,65102,65113,65124,65137 65161,65172,65203,65214,65224,65241,65252,65265,65276,65307,65320,65333,65344,65355,65366,65377 65421,65432,65445,65456,65467,65502,65513,65526,65534,65545,65553,65564,65575,65610,65623,65636 65657,65665,65676,65710,65716,65724,65735,65751,65762,65773,66004,66015,66026,66037,66052,66063 66102,66113,66126,66137,66150,66161,66172,66203,66214,66225,66236,66247,66262,66273,66304,66315 66337,66350,66362,66373,66404,66417,66430,66441,66447,66460,66471,66502,66513,66524,66535,66546 66565,66576,66607,66620,66631,66642,66653,66664,66675,66706,66717,66732,66743,66754,66765,67000 67026,67037,67050,67061,67072,67102,67112,67122,67136,67147,67160,67171,67202,67213,67224,67235 67256,67267,67277,67310,67316,67327,67335,67346,67357,67370,67376,67407,67426,67441,67452
13253	F1	2556,64525,64535
11177	F4	3656,66601,66611
12376	F8	3561,65402,65412
1366	FA	1354
1421	FB	1417
1424	FC	1341, 1402
1425	FD	1343, 1375
11760	GO	2604,66020,66030
11315	OR	2626,66463,66473
5047	RD	4756, 5001, 5027, 5032,70240
1666	SI	1625, 1647
13264	AND	2553,64514,64524
1134	ASG	1106, 1115
30	AWD	24
1700	BBT	76733,77005,77031
1560	BEX	1556, 1577
1676	BFS	76724,76731,76736,76752,76763,76776,77011
1702	BFW	76740,76765,76767,77024
12135	BIN	142,65651
7317	CPL	6473,65156,70240
3075	CTG	2743
1624	DCT	3145, 6420,11774,13076
7654	EOF	10323,70240
13674	EQP	65352,70240
1614	ERM	1541
1612	ERO	1543
1611	ERT	1537, 1547
1527	ERX	1535, 1536
11043	F13	2640,66735,66745
1401	FAA	1376
1402	FAB	1400
1430	FFU	1363, 1411
1431	FFV	1365, 1413
1432	FFW	1367, 1415
1427	FFX	1336, 1350, 1352, 1373
1422	FFY	1345
1426	FFZ	1337, 1344, 1347, 1371
12135	FIX	11174,11175,12621,12625,12775,12776,65643,65650,65651
3042	FRX	3037
7642	H00	166,11016,70136,70137
7643	H01	162,70135,70140

FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED
FAULTY REC. OMITTED

POST PROCESSOR ASSEMBLY DATA

7644	H02	162,70134,70141
7645	H03	163,70133,70142

```
7646    H04   163,70132,70143
7647    H05   164,70131,70144
7650    H06   164,70130,70145
7651    H07   165,70127,70146
7652    H10   165,70126,70147
7653    H11   171,70125,70150
7654    H12   167,65330,67475,70124,70151,70240
7655    H13 65374,70123,70152
7656    H14 66562,70122,70153
7657    H15 70121,70154
7660    H16 70120,70155
7661    H17 70117,70156
7662    H20 66543,70116,70157
7663    H21 70115,70160
7664    H22 70114,70161
7665    H23 70113,70162
7666    H24 70112,70163
7667    H25 70111,70164,77526,77527
7670    H26 70110,70165
7671    H27 70107,70166
7672    H30 70106,70167
7673    H31 70105,70170
7674    H32 70104,70171
7675    H33   172,66521,70103,70172
7676    H34   173,67034,70102,70173
7677    H35 65077,70101,70174
7700    H36 70100,70175
7701    H37 70077,70176
7702    H40   174,65236,70076,70177
7703    H41 70075,70200
7704    H42 70074,70201
7705    H43 70073,70202
7706    H44 70072,70203
7707    H45 70071,70204
7710    H46 70070,70205
7711    H47 70067,70206
7712    H50 70066,70207
7713    H51 70065,70210
7714    H52 70064,70211
7715    H53 65304,70063,70212
7716    H54 67253,70062,70213
7717    H55 65211,70061,70214
7720    H56 70060,70215
7721    H57 70057,70216
7722    H60 64626,70056,70217
7723    H61 67221,70055,70220
7724    H62 70054,70221
7725    H63 70053,70222
7726    H64 70052,70223
7727    H65 70051,70224
7730    H66 70050,70225
7731    H67 70047,70226
7732    H70 70046,70227
```

POST PROCESSOR ASSEMBLY DATA

7733	H71	70045,70230
7734	H72	170,65341,70012,70044,70231
7735	H73	65110,70043,70232
7736	H74	175,66211,70042,70233
7737	H75	70041,70234
7740	H76	70040,70235
7741	H77	70037,70236
11704	LAP	66074,66104
1711	LBT	103,76730,76732,76750
12375	MAX	66257,70240
1675	MFS	77014
12367	MIN	66270,70240
0	NIL	66353,70240
12	NP5	64466,64477,64511,64522,64533,64544,64552,64563,64574,64605,64616,64631,64642,64655,64666,64677
		64721,64732,64743,64754,64765,64776,65007,65020,65031,65042,65054,65067,65102,65113,65124,65137
		65161,65172,65203,65214,65224,65241,65252,65265,65276,65307,65320,65333,65344,65355,65366,65377
		65421,65432,65445,65456,65467,65502,65513,65526,65534,65545,65553,65564,65575,65610,65623,65636
		65657,65665,65676,65710,65716,65724,65735,65751,65762,65773,66004,66015,66026,66037,66052,66063
		66102,66113,66126,66137,66150,66161,66172,66203,66214,66225,66236,66247,66262,66273,66304,66315
		66337,66350,66362,66373,66404,66417,66430,66441,66447,66460,66471,66502,66513,66524,66535,66546
		66565,66576,66607,66620,66631,66642,66653,66664,66675,66706,66717,66732,66743,66754,66765,67000
		67026,67037,67050,67061,67072,67102,67112,67122,67136,67147,67160,67171,67202,67213,67224,67235
		67256,67267,67277,67310,67316,67327,67335,67346,67357,67370,67376,67407,67426,67441,67452
11334	OCT	150,66444,66451
1704	ORG	1150, 2771, 3007,76654,77220,77256,77452
12561	PJ1	65232
74	Q10	2437, 3116, 6571,10701
126	QD1	1334, 3505, 3526, 5446, 7713,11042,12666,12675,12711,12723,12754,13472,13674,14361,14525,14547
		77017,77034,77670
127	QD7	7215
130	QP5	1563,10411
122	QT1	11176,11430
123	QT2	3725,11201
124	QT4	3730
125	QT5	11173
10615	SET	11345,11346,67163,67173
35	STS	30
10446	SYM	155,64505,67076,67106,67116,67332,67337,67712,77320,77321,77507,77510
1677	TBT	76722,76757
1674	TFS	76640,76720,76761,76772
1701	TFW	76735,76754
1703	TPG	76653,76701,76745
12626	ADD1	64463,70240
12353	ADDP	66532,70240
13243	APP2	13151,70240
2527	ARG1	3210, 3225,10567,10576
2530	ARG2	7602, 7603, 7611, 7613, 7727, 7750,10570
2531	ARG3	3375, 3427, 3441, 3442, 7352, 7612, 7626, 7652, 7655, 7656, 7657,10130,10345,10552,10603,11341
		11717,11734,11765,12000,12011,12027,12045,12053,12570,12603,13135,13142,13153,13177,13220,13236
		13322,13333,13556,13601,13747,13750,14141,14177,14463,14623,14650,14712,15150,77460
2532	ARG4	3377, 3425, 3435, 3440, 3445,10370,11352,14713,15146
2533	ARG5	11354,14714,15144
2534	ARG6	14715,15142
2535	ARG7	14716,15140
2536	ARG8	14717,15136

POST PROCESSOR ASSEMBLY DATA

```
 2537   ARG9  14720,15134
13212   ATOM  64566,64576
 1602   BACD   1552, 1607
 1615   BACE   1605
 3012   BLKB   2767, 2775
 3013   BLKC   2776
 3016   BLKX   2766
 1153   BPSA   1145
13441   CARP  64652,66703,70240,77610
13451   CDRP  64663,70240,77464,77621
 2733   CNSX   2714
 3051   CNTA   3045
 3063   CNTB   3050
 3074   CNTM   2740, 2745, 2761, 2763, 3046, 3070
 3073   CNTS   2741, 2744, 2760, 3060,10355,10365,10424,10430
 3061   CNTX   3051
 3062   CNTY   3052
12647   COND   2572,14430,14431,65131,65141
 2714   CONS   1574, 3362, 3443, 5023, 6475, 6500, 6503, 6506, 7333, 7446, 7476,10017,10021,10024,10027,10071
               10275,10300,10401,10577,10740,10747,11070,11170,11614,11616,11621,11626,11631,11645,12005,12014
               13217,13503,13512,13531,13533,13752,13755,14414,14503,65145,77331,77334,77401,77403,77564,77571
               77624,77627,77712
 3334   COPY  65167,70240
 1727   CPPI   1557, 1570, 1576, 1731, 1740, 1747, 1773, 2000, 2025, 2070, 3203, 3206, 3207, 3255, 3271, 3273
               10371,10406,10416,14067,14112,14115,14116,14143,14155,14156,15031,15061,15067,15154,15163,76663
 2002   CSSI   1777, 2066,76664
 2707   CSWO   2705
 2712   CSWQ   2677, 2707, 3031, 3035
 2710   CSWX   2674
11161   CURC  11023,65231,70240
 1772   END0
 1770   END1  11214,11230,11244,13342,14267,14542,14642
 1766   END2   3347, 3621, 5006, 7366, 7433,12404,13312,13364,13542,14052,14176
 1764   END3   3477, 7510,13133,13624,13652,14231
 1762   END4   7535,13411,14360
 1760   END5   3333, 3371,14363
 1756   END6
 1754   END7
 1752   END8  10340
 1524   ERAC   2376
 1525   ERMQ   1533
13703   EVAL  10377,13140,13174,13406,13423,13434,13537,13553,13633,13661,14224,14445,14454,14543,65442,70240
13645   EVOR  66466,70240
12247   EXPR  10253,10254,13266,13267,13742,13743,14606,14607,65531,65536
12060   EXPT  65542,70240
 1310   FILA   1273
 1313   FILB   1274, 1277, 1301, 1315, 1320, 1322
 1312   FILC   1276, 1304, 1305, 1317, 1325, 1326
  142   FIXD   2764, 6567,11130,12115,12565
12713   FIXP  65654,70240
 2735   FREE   2135, 2364, 2715, 2723, 3232, 3237, 3302, 3306, 7411, 7415,14767,15011,15022,77016
 1250   GENX   1241
  166   H00A   6574,11100
  171   H11D  11100
  172   H33D   5045
```

POST PROCESSOR ASSEMBLY DATA

173	H34D	5044
174	H40D	1473,11100
170	H72A	11100
10366	H72H	67404,67417
175	H74D	5043
67455	HH00	70136
67456	HH01	70135
67457	HH02	70134
67460	HH03	70133
67461	HH04	70132
67462	HH05	70131
67463	HH06	70130
67464	HH07	70127
67465	HH10	70126
67466	HH11	70125
67467	HH12	70124
67476	HH13	70123
67502	HH14	70122
67506	HH15	70121
67512	HH16	70120
67516	HH17	70117
67522	HH20	70116
67526	HH21	70115
67532	HH22	70114
67536	HH23	70113
67542	HH24	70112
67546	HH25	70111
67552	HH26	70110
67562	HH27	70107
67566	HH30	70106
67572	HH31	70105
67576	HH32	70104
67602	HH33	70103
67606	HH34	70102
67612	HH35	70101
67616	HH36	70100
67622	HH37	70077
67626	HH40	70076
67632	HH41	70075
67636	HH42	70074
67642	HH43	70073
67646	HH44	70072
67652	HH45	70071
67656	HH46	70070
67662	HH47	70067
67666	HH50	70066
67672	HH51	70065
67676	HH52	70064
67702	HH53	70063
67706	HH54	70062
67714	HH55	70061
67720	HH56	70060
67724	HH57	70057
67730	HH60	70056
67734	HH61	70055

POST PROCESSOR ASSEMBLY DATA

```
67740   HH62 70054
67744   HH63 70053
67754   HH64 70052
67760   HH65 70051
67764   HH66 70050
67770   HH67 70047
67774   HH70 70046
70000   HH71 70045
70004   HH72 70044
70013   HH73 70043
70017   HH74 70042
70023   HH75 70041
70027   HH76 70040
70033   HH77 70037
65217   )PJ1 65232
13513   LABP 66060,70240
13477   LAMP 65732,70240
11647   LIST 11463,11464,66131,66141
11524   MAXP   160,12376,66254,66264
11513   MINP   160,12370,66265,66275
11163   MKNO  1152, 2765, 3026, 6601,10665,10674,10711,11270,11326,12001,12012,12033,12056,12117,12564,12647
              13103,66334,70240,77376
12361   MULT 67365,70240
70237   OBLB 70240,77040
  150   OCTD  3025, 6563,10673,11267,11325,77375
12760   ONEP 66455,70240
15216   OPEN  1117, 1165
10610   PACK 66477,70240
 7447   PAIR 13163,66510,70240
11625   PJ36   156,11211,66153,66163
11636   PJ37   157,11225,66142,66152
11603   PJ38   157,11241,66175,66205
11251   PLUS   161,12354,66527,66537
11155   PROG  2633,66623,66633
 7632   PROP 10554,70240
 4344   PUN2  4076,70240
11537   R69A  2621,66241,66253
11550   R69B  2615,66230,66240
 4755   READ  1204,10054,66740,70240
 1723   SAVE  3346, 3370, 3476, 3620, 5005, 7365, 7432, 7507, 7534,10337,11213,11227,11243,12403,13132,13311
              13363,13410,13541,13623,13651,14051,14175,14230,14266,14362,14541,14641
 1746   SAVI  1744
 1745   SAVJ  1742
 1751   SAVK  1743
 1776   SAVT  1724, 1737, 1772
 1774   SAVY  1723, 1736
  131   SBIT  7122, 7215
 1155   SEEK  1254,10050
 1475   SETA  1455, 1464
 1451   SETB  1445, 1471
 1476   SETC  1456, 1465
 1470   SETE  1452, 1461
 1467   SETG  1460
13575   SETP 67166,70240
10604   SETQ  2650,67174,67204
```

POST PROCESSOR ASSEMBLY DATA

```
 1446  SETX  1443
 1463  SETY  1453
10626  SRCH  2643,67152,67162
10504  STOP  10322,67274,67301
12655  SUB1  67305,70240
10465  SUBR   153,10251,10252,13264,13265,13736,13737,14600,14601,64461,64472,64526,64537,64556,64567,64600
             64635,64650,64661,64672,64703,64714,64725,64736,64747,64760,64771,65002,65013,65024,65035,65062
             65143,65154,65165,65176,65245,65260,65271,65313,65350,65361,65403,65414,65425,65440,65451,65462
             65506,65521,65540,65557,65570,65603,65616,65631,65652,65671,65755,65766,65777,66010,66032,66045
             66106,66121,66165,66220,66231,66242,66277,66310,66321,66332,66343,66355,66366,66377,66412,66423
             66475,66506,66571,66602,66613,66635,66646,66657,66670,66712,66725,66736,66747,66760,66773,67006
             67043,67054,67065,67131,67142,67153,67164,67206,67230,67262,67303,67313,67320,67322,67341,67352
             67434,67445,77074,77075,77511,77512
  176  WBUF  1132
 1673  XORG  2773, 3011,77222,77260,77454
 1505  XPLA  1500
 2673  XPLE  1502, 1504, 1514, 1517
 1507  XPLG  4100
 1523  XPLS  1510, 1511
 1520  XPLX  1507
 2526  ALIST  1340, 1342,10111,13312,13314,13364,13366,13557,14052,14053,14176,14200,14251,14622,14647,15203
  132  AMASK   113, 2756, 3705,10354,10423,11574,11604,14523,77322,77354,77520
 1115  APEND  1377, 1420
13110  APPLY  3447, 7662,10120,10606,13242,13277,13340,13343,13766,14163,14272,14626,14652,64541,70240
 7632  APROP 66637,70240
13231  APVAL    35,  141,14000,14001,64547,64554,64555,64624,65046,65075,65106,65207,65220,65234,65302,65326
             65372,65702,65743,66207,66434,66517,66541,66560,67032,67217,67251,67402,67553,67745
 2540  ARG10 14721,15132
 2541  ARG11 14722,15130
 2542  ARG12 14723,15126
 2543  ARG13 14724,15124
 2544  ARG14 14725,15122
 2545  ARG15 14726,15120
 2546  ARG16 14727,15116
 2547  ARG17 14730,15114
 2550  ARG18 14731,15112
 2551  ARG19 14732,15110
 2552  ARG20  7741,14733,15106
13223  ARRAY  2103, 2104,11677,64555,64565
 1106  ASIGN  1401, 4415
13457  ATOMP 64571,70240
 2747  AWHOA  2742
 1604  BACER  1545
 1600  BACTD  1560
 3020  BLKBB  3001, 3015
77740  BUTCH 77057,77741,77742
11162  CHACT 10777,11017,11021,65051,70240
10753  CLEAR 65064,70240
15222  CLOSE  1157, 1263, 1356, 1405,10145
 2726  CNTR1  2732, 2757, 3047, 3056, 3067, 3243, 3247, 3312, 3316,10353,10363,10422,10426,14773,14777
 3065  CNTST  2755, 2762, 3055
 2674  CONSW  7331,10015,10736,10745,11165,11624,77116,77127,77150,77323
12614  COPYN  2567,65164,65174
 3044  COUNT 65200
11160  CURC1 11024,65221,70240
```

POST PROCESSOR ASSEMBLY DATA

```
 3076  DECON  2245, 2250, 2253,70240
11037  DIGIT 65262,70240
  133  DMASK  3367, 6502, 7447, 7562, 7600, 7713, 7725, 7771,10032,10517,11167,13457,77274,77327,77552,77607
 3450  EQUAL  3500, 7356,65405,70240,77555
 1526  ERIND  1534
 1530  ERROR  1670, 1671, 1720, 1721, 2004, 2005, 2355, 2400, 2401, 2751, 2752, 3023, 3027, 3700, 3703, 4772
               5105, 5106, 5435, 6545, 6551, 7477, 7500, 7502, 7503, 7743, 7744,10641,10642,10714,10715,11436
              11655,11656,12604,12605,13346,13347,13572,13573,13613,13614,14037,14040,14242,14243,14510,14511
              14743,77207,77210,77504,77505,77637,77640
13617  EVAND 64517,70240
13406  EVCON 65134,70240
14225  EVLIS 11215,11231,11245,12405,65453,66134,70240
 1271  FENDA  1260
12230  FEXPR 10260,10261,13744,13745,14610,14611,65550,65555
 1336  FILE1 65561
 1256  FILSA  1252
12116  FLOAT   143,11177,11200,12622,12777,13000,65662,65667
 3037  FROUT  2717, 3234, 3304, 7413,14771
12065  FSUBR   144,10256,10257,13740,13741,14602,14603,64515,65132,65713,65720,65730,66021,66056,66132,66143
              66176,66255,66266,66464,66530,66624,66701,67175,67363,77076,77077,77513,77514
 2675  FWLOR  3036
 1176  GCGAG 65757
 1231  GETBA  1216
 1240  GETBB  1224, 1230, 1233, 1236
 1233  GETBD  1227
 1225  GETBG  1237
67412  HH72H 67417
14361  INTER 66626,70240
12466  .E001 65312,65324
65316  .E002 65314,65316
65322  .E003 65320,65321
65321  .E004 65320
12453  .E006 65325,65335
65331  .E007 65327,65331
65334  .E008 65333
12442  .E010 65336,65346
65342  .E011 65340,65342
65345  .E012 65344
12431  .E014 65347,65357
65353  .E015 65351,65353
65356  .E016 65355
12420  .E018 65360,65370
65364  .E019 65362,65364
65367  .E020 65366
12407  .E022 65371,65401
65375  .E023 65373,65375
65400  .E024 65377
65406  .E026 65404,65406
65411  .E027 65410
12365  .E029 65413,65423
65417  .E030 65415,65417
65422  .E031 65421
65430  .E033 65426,65430
65434  .E034 65432,65433
65433  .E035 65432
```

POST PROCESSOR ASSEMBLY DATA

```
12341   .E037 65437,65447
65443   .E038 65441,65443
65446   .E039 65445
65454   .E041 65452,65454
65457   .E042 65456
12317   .E044 65461,65473
65465   .E045 65463,65465
65471   .E046 65467,65470
65470   .E047 65467
12304   .E049 65474,65504
65500   .E050 65476,65500
65503   .E051 65502
12273   .E053 65505,65517
65511   .E054 65507,65511
65515   .E055 65513,65514
65514   .E056 65513
12260   .E058 65520,65530
65524   .E059 65522,65524
65527   .E060 65526
65535   .E062 65534
12241   .E064 65537,65547
65543   .E065 65541,65543
65546   .E066 65545
65554   .E068 65553
12222   .E070 65556,65566
65562   .E071 65560,65562
65565   .E072 65564
12211   .E074 65567,65601
65573   .E075 65571,65573
65577   .E076 65575,65576
65576   .E077 65575
12176   .E079 65602,65614
65606   .E080 65604,65606
65612   .E081 65610,65611
65611   .E082 65610
12163   .E084 65615,65627
65621   .E085 65617,65621
65625   .E086 65623,65624
65624   .E087 65623
12150   .E089 65630,65642
65634   .E090 65632,65634
65640   .E091 65636,65637
65637   .E092 65636
65647   .E094 65646
12127   .E096 65651,65661
65655   .E097 65653,65655
65660   .E098 65657
65666   .E100 65665
12110   .E102 65670,65700
65674   .E103 65672,65674
65677   .E104 65676
12077   .E106 65701,65712
65706   .E107 65703,65706
65705   .E108 65704
65711   .E109 65710
```

POST PROCESSOR ASSEMBLY DATA

```
65717  .E111 65716
65725  .E113 65724
12051  .E115 65727,65741
65733  .E116 65731,65733
65737  .E117 65735,65736
65736  .E118 65735
12036  .E120 65742,65753
65747  .E121 65744,65747
65746  .E122 65745
65752  .E123 65751
12024  .E125 65754,65764
65760  .E126 65756,65760
65763  .E127 65762
12013  .E129 65765,65775
65771  .E130 65767,65771
65774  .E131 65773
12002  .E133 65776,66006
66002  .E134 66000,66002
66005  .E135 66004
11771  .E137 66007,66017
66013  .E138 66011,66013
66016  .E139 66015
66024  .E141 66022,66024
66027  .E142 66026
11747  .E144 66031,66043
66035  .E145 66033,66035
66041  .E146 66037,66040
66040  .E147 66037
11734  .E149 66044,66054
66050  .E150 66046,66050
66053  .E151 66052
66061  .E153 66057,66061
66064  .E154 66063
66072  .E156 66071
66100  .E158 66076,66100
66103  .E159 66102
11673  .E161 66105,66117
66111  .E162 66107,66111
66115  .E163 66113,66114
66114  .E164 66113
11660  .E166 66120,66130
66124  .E167 66122,66124
66127  .E168 66126
66135  .E170 66133,66135
66140  .E171 66137
66146  .E173 66144,66146
66151  .E174 66150
66157  .E176 66155,66157
66162  .E177 66161
11614  .E179 66164,66174
66170  .E180 66166,66170
66173  .E181 66172
66201  .E183 66177,66201
66204  .E184 66203
11572  .E186 66206,66216
```

POST PROCESSOR ASSEMBLY DATA

```
66212  .E187 66210,66212
66215  .E188 66214
13320  ..001 64460,64470
64464  ..002 64462,64464
64467  ..003 64466
13307  ..005 64471,64503
64475  ..006 64473,64475
64501  ..007 64477,64500
64500  ..008 64477
13274  ..010 64504,64513
64507  ..011 64506,64507
64512  ..012 64511
64520  ..014 64516,64520
64523  ..015 64522
64531  ..017 64527,64531
64534  ..018 64533
13242  ..020 64536,64546
64542  ..021 64540,64542
64545  ..022 64544
64553  ..024 64552
64561  ..026 64557,64561
64564  ..027 64563
64572  ..029 64570,64572
64575  ..030 64574
13201  ..032 64577,64607
64603  ..033 64601,64603
64606  ..034 64605
13170  ..036 64610,64622
64614  ..037 64612,64614
64620  ..038 64616,64617
64617  ..039 64616
13155  ..041 64623,64633
64627  ..042 64625,64627
64632  ..043 64631
13144  ..045 64634,64646
64640  ..046 64636,64640
64644  ..047 64642,64643
64643  ..048 64642
13131  ..050 64647,64657
64653  ..051 64651,64653
64656  ..052 64655
13120  ..054 64660,64670
64664  ..055 64662,64664
64667  ..056 64666
13107  ..058 64671,64701
64675  ..059 64673,64675
64700  ..060 64677
13076  ..062 64702,64712
64706  ..063 64704,64706
64711  ..064 64710
13065  ..066 64713,64723
64717  ..067 64715,64717
64722  ..068 64721
13054  ..070 64724,64734
64730  ..071 64726,64730
```

POST PROCESSOR ASSEMBLY DATA

```
64733   ..072 64732
13043   ..074 64735,64745
64741   ..075 64737,64741
64744   ..076 64743
13032   ..078 64746,64756
64752   ..079 64750,64752
64755   ..080 64754
13021   ..082 64757,64767
64763   ..083 64761,64763
64766   ..084 64765
13010   ..086 64770,65000
64774   ..087 64772,64774
64777   ..088 64776
12777   ..090 65001,65011
65005   ..091 65003,65005
65010   ..092 65007
12766   ..094 65012,65022
65016   ..095 65014,65016
65021   ..096 65020
12755   ..098 65023,65033
65027   ..099 65025,65027
65032   ..100 65031
12744   ..102 65034,65044
65040   ..103 65036,65040
65043   ..104 65042
12733   ..106 65045,65060
65052   ..107 65047,65052
65051   ..108 65050
65056   ..109 65054,65055
65055   ..110 65054
12717   ..112 65061,65073
65065   ..113 65063,65065
65071   ..114 65067,65070
65070   ..115 65067
12704   ..117 65074,65104
65100   ..118 65076,65100
65103   ..119 65102
12673   ..121 65105,65115
65111   ..122 65107,65111
65114   ..123 65113
12662   ..125 65116,65130
65122   ..126 65120,65122
65126   ..127 65124,65125
65125   ..128 65124
65135   ..130 65133,65135
65140   ..131 65137
12636   ..133 65142,65152
65146   ..134 65144,65146
65151   ..135 65150
12625   ..137 65153,65163
65157   ..138 65155,65157
65162   ..139 65161
65170   ..141 65166,65170
65173   ..142 65172
12603   ..144 65175,65205
```

POST PROCESSOR ASSEMBLY DATA

```
65201   ..145 65177,65201
65204   ..146 65203
12572   ..148 65206,65216
65212   ..149 65210,65212
65215   ..150 65214
65226   ..152 65224,65225
65225   ..153 65224
12545   ..155 65233,65243
65237   ..156 65235,65237
65242   ..157 65241
12534   ..159 65244,65256
65250   ..160 65246,65250
65254   ..161 65252,65253
65253   ..162 65252
12521   ..164 65257,65267
65263   ..165 65261,65263
65266   ..166 65265
12510   ..168 65270,65300
65274   ..169 65272,65274
65277   ..170 65276
12477   ..172 65301,65311
65305   ..173 65303,65305
65310   ..174 65307
66223   .M001 66221,66223
66226   .M002 66225
66234   .M004 66232,66234
66237   .M005 66236
66245   .M007 66243,66245
66251   .M008 66247,66250
66250   .M009 66247
66260   .M011 66256,66260
66263   .M012 66262
66271   .M014 66267,66271
66274   .M015 66273
11502   .M017 66276,66306
66302   .M018 66300,66302
66305   .M019 66304
11471   .M021 66307,66317
66313   .M022 66311,66313
66316   .M023 66315
11460   .M025 66320,66330
66324   .M026 66322,66324
66327   .M027 66326
11447   .M029 66331,66341
66335   .M030 66333,66335
66340   .M031 66337
11436   .M033 66342,66352
66346   .M034 66344,66346
66351   .M035 66350
11424   .M037 66354,66364
66360   .M038 66356,66360
66363   .M039 66362
11413   .M041 66365,66375
66371   .M042 66367,66371
66374   .M043 66373
```

POST PROCESSOR ASSEMBLY DATA

```
11402  .M045 66376,66410
66402  .M046 66400,66402
66406  .M047 66404,66405
66405  .M048 66404
11367  .M050 66411,66421
66415  .M051 66413,66415
66420  .M052 66417
11356  .M054 66422,66432
66426  .M055 66424,66426
66431  .M056 66430
11345  .M058 66433,66443
66437  .M059 66435,66437
66442  .M060 66441
66450  .M062 66447
11326  .M064 66452,66462
66456  .M065 66454,66456
66461  .M066 66460
66467  .M068 66465,66467
66472  .M069 66471
11304  .M071 66474,66504
66500  .M072 66476,66500
66503  .M073 66502
11273  .M075 66505,66515
66511  .M076 66507,66511
66514  .M077 66513
11262  .M079 66516,66526
66522  .M080 66520,66522
66525  .M081 66524
66533  .M083 66531,66533
66536  .M084 66535
11240  .M086 66540,66550
66544  .M087 66542,66544
66547  .M088 66546
66555  .M090 66554
11221  .M092 66557,66567
66563  .M093 66561,66563
66566  .M094 66565
11210  .M096 66570,66600
66574  .M097 66572,66574
66577  .M098 66576
66605  .M100 66603,66605
66610  .M101 66607
11166  .M103 66612,66622
66616  .M104 66614,66616
66621  .M105 66620
66627  .M107 66625,66627
66632  .M108 66631
11144  .M110 66634,66644
66640  .M111 66636,66640
66643  .M112 66642
11133  .M114 66645,66655
66651  .M115 66647,66651
66654  .M116 66653
11122  .M118 66656,66666
66662  .M119 66660,66662
```

POST PROCESSOR ASSEMBLY DATA

```
66665   .M120 66664
11111   .M122 66667,66677
66673   .M123 66671,66673
66676   .M124 66675
66704   .M126 66702,66704
66707   .M127 66706
11067   .M129 66711,66723
66715   .M130 66713,66715
66721   .M131 66717,66720
66720   .M132 66717
11054   .M134 66724,66734
66730   .M135 66726,66730
66733   .M136 66732
66741   .M138 66737,66741
66744   .M139 66743
11032   .M141 66746,66756
66752   .M142 66750,66752
66755   .M143 66754
11021   .M145 66757,66771
66763   .M146 66761,66763
66767   .M147 66765,66766
66766   .M148 66765
11006   .M150 66772,67004
66776   .M151 66774,66776
67002   .M152 67000,67001
67001   .M153 67000
10773   .M155 67005,67017
67011   .M156 67007,67011
67015   .M157 67013,67014
67014   .M158 67013
10760   .M160 67020,67030
67024   .M161 67022,67024
67027   .M162 67026
10747   .M164 67031,67041
67035   .M165 67033,67035
67040   .M166 67037
10736   .M168 67042,67052
67046   .M169 67044,67046
67051   .M170 67050
10725   .M172 67053,67063
67057   .M173 67055,67057
67062   .M174 67061
10714   .M176 67064,67074
67070   .M177 67066,67070
67073   .M178 67072
10703   .M180 67075,67104
67100   .M181 67077,67100
67103   .M182 67102
10673   .M184 67105,67114
67110   .M185 67107,67110
67113   .M186 67112
10663   .M188 67115,67126
67120   .M189 67117,67120
67124   .M190 67122,67123
67123   .M191 67122
```

POST PROCESSOR ASSEMBLY DATA

```
10650  .S001 67130,67140
67134  .S002 67132,67134
67137  .S003 67136
10637  .S005 67141,67151
67145  .S006 67143,67145
67150  .S007 67147
67156  .S009 67154,67156
67161  .S010 67160
67167  .S012 67165,67167
67172  .S013 67171
67200  .S015 67176,67200
67203  .S016 67202
10573  .S018 67205,67215
67211  .S019 67207,67211
67214  .S020 67213
10562  .S022 67216,67226
67222  .S023 67220,67222
67225  .S024 67224
10551  .S026 67227,67237
67233  .S027 67231,67233
67236  .S028 67235
67245  .S030 67243,67244
67244  .S031 67243
10530  .S033 67250,67260
67254  .S034 67252,67254
67257  .S035 67256
10517  .S037 67261,67273
67265  .S038 67263,67265
67271  .S039 67267,67270
67270  .S040 67267
67300  .S042 67277
10476  .S044 67302,67312
67306  .S045 67304,67306
67311  .S046 67310
67317  .S048 67316
10457  .S050 67321,67331
67325  .S051 67323,67325
67330  .S052 67327
67336  .S054 67335
10440  .S056 67340,67350
67344  .S057 67342,67344
67347  .S058 67346
10427  .S060 67351,67361
67355  .S061 67353,67355
67360  .S062 67357
67366  .S064 67364,67366
67371  .S065 67370
67377  .S067 67376
10377  .S069 67401,67411
67405  .S070 67403,67405
67410  .S071 67407
10360  .S073 67420,67432
67424  .S074 67422,67424
67430  .S075 67426,67427
67427  .S076 67426
```

POST PROCESSOR ASSEMBLY DATA

```
10345  .S078 67433,67443
67437  .S079 67435,67437
67442  .S080 67441
10334  .S082 67444,67454
67450  .S083 67446,67450
67453  .S084 67452
65224  )PJ1A 65223
65230  )PJ1B 65223
11723  LABEL   146,66055,66065
11207  LOGOR 66156,70240
10721  MKNAM 10637,66323,70240
 7562  NCONC  1516, 7561,11640,13165,14400,66345,70240
 2004  NOPDL  1730,15157
13472  NULLP 66357,66370,70240
 6607  NUMBR  6556,70240
10647  NUMOB 66414,70240
 6531  NUTRN  5221,70240
 1056  OUTSW  1353, 1366, 1370, 1416, 4412, 4414,10153
  134  PMASK  1562,10410
11227  PNAME    31,   40,  151, 3711, 3712, 6444, 6445,64464,64475,64507,64520,64531,64542,64550,64561,64572
                64614,64627,64640,64653,64664,64675,64706,64717,64730,64741,64752,64763,64774,65005,65016,65027
                65052,65065,65100,65111,65122,65135,65146,65157,65170,65201,65212,65222,65237,65250,65263,65274
                65316,65331,65342,65353,65364,65375,65406,65417,65430,65443,65454,65465,65500,65511,65524,65532
                65551,65562,65573,65606,65621,65634,65644,65655,65663,65674,65706,65714,65722,65733,65747,65760
                66002,66013,66024,66035,66050,66061,66067,66100,66111,66124,66135,66146,66157,66170,66201,66212
                66234,66245,66260,66271,66302,66313,66324,66335,66346,66360,66371,66402,66415,66426,66437,66445
                66467,66500,66511,66522,66533,66544,66551,66552,66556,66563,66574,66605,66616,66627,66640,66651
                66673,66704,66715,66730,66741,66752,66763,66776,67011,67024,67035,67046,67057,67070,67100,67110
                67134,67145,67156,67167,67200,67211,67222,67233,67241,67254,67265,67275,67306,67314,67325,67333
                67355,67366,67374,67405,67413,67424,67437,67450,67467,67476,67502,67506,67512,67516,67522,67526
                67536,67542,67546,67556,67562,67566,67572,67576,67602,67606,67612,67616,67622,67626,67632,67636
                67646,67652,67656,67662,67666,67672,67676,67702,67706,67714,67720,67724,67730,67734,67740,67750
                67760,67764,67770,67774,70000,70004,70013,70017,70023,70027,70033,77100,77101
 3577  PRIN0  4337,70240
 3665  PRIN1  3611, 3637,66573,70240
 4076  PRIN2  3171, 3175, 3576, 3616, 3635, 3643, 3645, 3652, 3744, 3761, 4007, 4023, 4042, 4056, 4333, 4334
                4443, 4532, 4534, 4704, 4707,14310,14342,14344,70240
 3564  PRINT  1501, 1550,10137,10242,14313,14322,14346,14351,66604,70240,77050,77233
 4327  PUNCH  1403,66650,70240
  117  QO1Q9 13045
  155  QSYMD 77203
11100  QUOTE    152,66700,66710,77535,77536
 1202  RDFLX 66727
 1436  SAVBK 67144
 1441  SAVSY  1436
 1443  SETBK    102,67210
13540  SETQP 67177,70240
 2756  SPEAK 67232
  153  SUBRD 11700
 7350  SUBST 67324,70240
10443  TERA2 10450
10416  TIMES   161,12362,67362,67372
10405  TRACE 13262,13263,13734,13735,14604,14605,67373,67400
13012  UNFIX 11732,11741,12110,12130,12514,70240
 6521  VALUE  6406, 6411, 6451, 6472, 6535,70240
```

POST PROCESSOR ASSEMBLY DATA

```
15206   WRFLX  1057, 1542, 1604, 2255, 2362, 4157, 4246, 5101, 5432, 6546,10134,10165,10237
 1522   XPLSW  1477, 1503, 4077, 4244
12734   ZEROP  3554,67447,70240
77744   ZEROX 77740
13405 A     A 13252,13313,13324,13332,13365
13403 A     F 13254,13323,13345
13404 A    AL 13253,13305,13334,13360
13414 A    E1 13436
13435 A    E2 13425
13432 A    E3 13440
13300 A    R2 13265
14327 A   AGA 14303,14312,14336,14345
14331 A   AGM 14307
14333 A   AGO 14343
14330 A   AGQ 14304,14314,14337,14347
13376 A   APA 13352,13361
 2556 A   AS1  7431, 7445
 2525 A   CSV 13247,13317,13342,13354,13372,13374,14050,14061,14174,14205,14256,14260
13402 A   FAS 13321
 2632 A   LIS  7450, 7453, 7462, 7466
14315 A   PLL 14323
14325 A   PRX 14301
14324 A   PRY 14302,14315
13345 A   R33 13402
 2631 A   TEM  7461, 7471
14350 A   VAX 14335
13243 A  APP2 70240
13147 A  ASP1 13117
13152 A  ASP2 13123
13175 A  ASP3 13131
13225 A  ASP4 13126
 2560 A  ASS1 13110,13114,13145,13150,13173,13223,13241,13353,13355
 2562 A  ASSA 13133,13136,13141
 2561 A  ASSL 13134,13143
 2563 A  AST1 13112,13137,13147,13152,13175,13211,13222,13225,13231,13237
 2564 A  AST2 13176,13221
 2565 A  AST3 13154,13164,13200,13216
 2566 A  AST4 13160,13167,13204,13213
13401 A  ATS1 13243,13245,13246,13276,13337
 2557 A  CWR1  7433, 7436, 7442
  133 A  DECM  7437, 7447
 2572 A  ECS1 13407,13433
 2573 A  ECS2 13412,13424
 2574 A  ECS3 13415,13435
 2575 A  ECS4 13411,13420,13426
14214 A  ELP1 14135,14233
14223 A  ELT1 14215
13771 A  EVIN 13712
14112 A  EV27 13727,14213
13703 A  EVAL 70240
14360 A  EVCM 14127
14356 A  EVD1 14017,14033,14076,14077
 2603 A  EVD2 13746,13757
14027 A  EVI1 14014
14030 A  EVI2 14016
```

POST PROCESSOR ASSEMBLY DATA

```
14106 A EVI3 14072
14107 A EVI4 14074
14021 A EVL1 14027,14030
14100 A EVL2 14106,14107
13774 A EVP1 13714,14000,14001
13730 A EVP2 13744,13745,13770
 2576 A EVS1 13703,13716,13765,13772,14010,14034,14047,14117,14145,14162,14173,14225,14227,14236,14263,14267
             14277
 2600 A EVSA 14123,14133,14140,14151,14222,14231,14232
 2577 A EVSE 14121,14132,14142,14147
14355 A EVT1 14046,14057,14075,14110,14137,14157,14172,14203,14240,14253,14270
14354 A EVTA 14012,14020
14352 A EVTE 13705,13771,14013,14036,14241
 7457 A FARG  7451
 7447 A PAIR 70240
 7477 A PERF  7454
 7502 A PERS  7463
14334 A VALV 14341
13370 AAPEXC 13344
 7431 AAPNP1  7426
13110 AAPPLY 13146,13224,70240
13321 AAPSAL 13257
13256 AAPSES 13266,13267,13362
13351 AAPTRK 13263
13377 AAPTRT 13255,13274,13307,13335,13351
13363 AAPTSB 13310
13341 AAPTXP 13275,13336
14301 AARGOF 13357,14250,14265,14643
  145 AASFUN 13124,13243
  146 AASLBL 13127,13243
  147 AASLMD 13121,13243
13400 ACWADR 13304,13315,13367
10763 ADVANC 64474
14162 AEVAPG 14167
13406 AEVCON 70240
14164 AEVOCO 14161
14225 AEVLIS 70240
14357 AEVLNS 13710,13711,13721,13722
14012 AEVP11 13775
14036 AEVP12 14021
14042 AEVP22 13741
14063 AEVP23 13743
14071 AEVP25 13731
14240 AEVP26 13723,14100
14207 AEVP27 13737
14114 AEVP28 14070
  126 AEVQD1 14015,14073,14361
14353 AEVTAE 13715,14066,14071,14111,14131
 2602 AEVTDE 13717,13754,14056,14134,14247,14252,14262,14271
14245 AEVTFS 14055
 2601 AEVTRK 13724,13725,13763,13767,14054,14125,14153,14160,14164,14166,14170,14212,14360
13767 AEVTRT 13735
14262 AEVTXP 13764,14165
 7475 AFARGX  7460
 7455 APAIRX  7447
```

POST PROCESSOR ASSEMBLY DATA

```
 7426 APPEND  7440,10076,13506,64530
 7663 APSSOC 67133,70240
13231 APVALI 64555,67473,70010
 2736 ARREST  2730, 3245, 3314,14775
11332 ARYGET 11661
11370 ARYGTX 11660
 2515 ARYLIS  2074,11644,11646
11441 ARYMAK 64560
 7755 ATTRIB 64602
14335 AVALOF 13375,14261,14300,14660
   76 B ZERO    77,  105,14565
 1610 BACACT  1544, 1546, 1602
 1433 BACKTR  1601,64613
12120 BASFLT 12103
 4315 BCDADJ  4326,10012
  141 BCONAT  2507
 2514 BEGBLK  2513
 2766 BLOCKR 11546
 1712 BPLACE  1716, 2770, 2772, 3002, 3004, 3006, 3010,15051,76641,77217,77221,77237,77241,77255,77257,77451
             77543,77567,77572,77673,77707,77713
 1145 BPSREM 64637,77047
12121 BSFLTA 12111
64457 BUCKET  6422, 6511,64460
 6523 BUCKNO  6416
15220 BUFFER  1127
 7301 BUFFRE  6542, 6554, 6606
 6405 BUKSRT 77157
10607 C    CA 10573,10602
10602 C    CD 10575
10575 C    CL 10601
10604 C    CX 10565
10605 C    CY 10566
77507 C    PA 77502
77702 C    PH 77667
77701 C    PL 77671
77703 C    PX 77665,77716,77720
 7317 C   CPI 70240
 7343 C   CPA  7324
 7346 C   CPE  7320, 7321, 7336, 7340
 2571 C   CPF  7323, 7334, 7335, 7343
 7325 C   CPL  7342
 7347 C   CPT  7326, 7341
 7344 C   CPX  7317
10547 C   GET  1220,15053,66001,77204,77303,77611
15026 C   LAN 14772,15000,15016,15020
77162 C   LAP  3021,66077
77342 C   LAX 77162,77163
77445 C   LEM 77653
14740 C   LER 14554,14563
15013 C   LFX 15005,15023
14736 C   LNN 14705
14774 C   LSC 14757
15017 C   LSE 14766
14760 C   LSN 14756
15001 C   LSP 14765,15010
```

POST PROCESSOR ASSEMBLY DATA

```
77212 C  LSQ 77201,77205
15006 C  LSR 15025
14754 C  LST 14707,15201
15015 C  LX2 14754
15200 C  MOV 67107
77561 C  NON 77545
77604 C  NQT 77535,77536
77535 C  NTE 77526,77527
77706 C  PMK 77676
77675 C  PNL 77701,77702
77665 C  PRO 77532,77632
77721 C  PTR 77666,77704,77711
77733 C  REM 77645,77647,77656,77661
77632 C  SPP 77612
77731 C  SUM 77643,77654,77655
15174 C  TAB 77165,77231,77336,77340,77402,77404,77461
77600 C  TRP 77603,77634
15061 C  UNF 15037
15062 C  UNX 15027
77416 C AINS 77407
77434 C AINX 77416,77441
15203 C ALST 64506
77406 C AMBL 77366,77371
77734 C ERCC 77476,77503
77663 C FELX 77436,77455,77470,77521,77525,77534,77601,77644,77652
77515 C FIND 77512
77522 C FINX 77510
10564 C GETL 10550,10553
10561 C GETX 10547,10563
77730 C HOLD 77547,77557,77565,77577,77602
77217 C INBP 77172,77175
77323 C IND2 77320,77321
77347 C INDC 77276,77300,77313,77316
77722 C INST 77341,77365,77400,77440,77443
77351 C JUST 77224,77422,77431
15175 C LCOM 77615,77623
77336 C LEND 77253,77267,77315
15020 C LFIX 15007
14552 C LINK    75
15171 C LIST 77164,77245,77261,77337
14747 C LNAC 14632,14644
14746 C LNFN 14615,14624,14646
14615 C LNGN 14630
 2610 C LNKA 14552,14662,14700,14710,14741
 2611 C LNKB 14553,14663,14701,14711
14751 C LNKC 14564,14625,14637,14642,14655,14657,14676
14752 C LNKD 14560,14570,14627,14636,14704
14710 C LNKP 14706
14574 C LNLP 14610,14611,14645
14734 C LNLX 14703,14737
14627 C LNNF 14573,14575
14631 C LNTR 14605
77360 C LOP1 77373,77405,77413
77477 C LOP2 77513,77514
77646 C LOPL 77657
```

POST PROCESSOR ASSEMBLY DATA

```
77735 C LSAC 77457
15201 C LSTR 67077
77727 C MODE 77176,77223,77234,77252
15153 C MOVD 15103,15105
15066 C MOVE 15200
15073 C MOVY 15066,15152
77345 C NAME 77270,77301,77325
77526 C NATM 77447
77471 C NTAB 77463
77602 C ONQT 77556
77357 C PASS 77230,77251
77414 C PAUX 77357,77361
15176 C PROS 15050,77672,77706,77715
77723 C REST 77170,77247,77360,77363
15202 C RTRN 67117
77724 C STAR 67713,77225,77236,77244,77254,77374,77410,77412,77420,77423,77427,77433
15161 C STRW 15076,15155
15162 C TXLW 15072,15075,15077
77346 C TYPE 77273,77302,77317,77333
15036 C UNLP 15042,15047,15054,15060
15065 C UNSQ 15030,15063
15037 C UNTL 15035
10450 CAAARX 64740
10463 CAADRX 64751
10504 CAARXX 64674
10470 CADARX 64762,77275
10477 CADDRX 64773
10506 CADRXX 64705,77531,77617,77631
77436 CAFELD 77417,77421,77424,77430
10510 CDAARX 65004
10522 CDADRX 65015
10543 CDARXX 64716
10527 CDDARX 65026
10536 CDDDRX 65037
10545 CDDRXX 64727
10563 CFCN31 10551
77635 CFDLST 77604,77605
77546 CFLOOP 77560
15242 CHNCOM 10150
 5370 CILMAR  5277, 5330, 5337, 5341, 5345, 5463
77225 CLAPP1 77216
77243 CLAPP2 77235
77456 CLAPP3 77450
 3022 CLBPTP  1146, 3003,76660,77240
14703 CLNARS 14620,14633
14664 CLNDIS 14661
14612 CLNEXP 14607
14750 CLNRGL 14634,14651
14666 CLNSBR 14601,14603
14653 CLNTEN 14665
14646 CLNTEX 14617
14745 CLNTRS 14567,14616,14631,14672
14661 CLNTSB 14673
14753 CLNTSX 14674,14675
77316 CMKIND 77304
```

POST PROCESSOR ASSEMBLY DATA

```
77465 CNEVAL 77475
77732 CNOCUR 77437,77635,77642,77660
77060 CNSFWL 77041
  136 CNTMSK 15102
 2754 CNTPTR  2750
   44 COMMON  1327,   44, 4544, 4552, 4555, 4560, 4570, 4573, 4577, 4604, 4606, 4616, 4617, 4621, 4623, 4625
              4635, 4646, 4651, 4670,   44,12140
10565 COMPAT  3324, 3330, 3406, 3417, 7520, 7545
 1200 COMPRI 65121
   77 COMPWD  1200,77232
76651 COMPZR 76631
77726 CPASWD 77227,77250,77372,77406
77350 CPATCH 77311
15171 CPROBE  2511,15171
15177 CPROEN  2511,15200
15172 CQTLST 77542,77566,77574
   74 CRITWN  2206, 2437
77725 CSTART 77226,77243,77277,77310
15027 CUNWND  1567,10415
 2614 D    F  7510, 7511, 7513, 7515
 2613 D    L  7512, 7522
 2612 D  RET  7506, 7530
 7512 D MCPR  7526
15214 DELFIL  1110, 1303
12042 DIFFER  3551,65247
 1057 DISKER  1125, 1133, 1173, 1266, 1410, 4424, 5326
 1072 DISKNG  1060
11750 DIVIDE 65273
15210 DORMNT  1061
 2020 E    A 76751
 2021 E    B 76753
 2136 E    F 76762
 2143 E    G 76777
 2156 E    H  2170,76766
 2157 E    I 76755
 2476 E  BIT  2347, 2477
77134 E  CMK 77124
 2521 E  CNX 77123,77132,77140,77160
 3115 E  DE1  3124
 3131 E  DE2  3123
 3114 E  DE4  3130
 3165 E  DE5  3147
 3106 E  DE7  3101
 3155 E  DEJ  3151
 3141 E  DEQ  3133
 3144 E  DEV  3132
 3176 E  DEY  3173
 2426 E  FSC  2133, 2145, 2216, 2247,65705
 2425 E  FWC  2155, 2176, 2210, 2244, 2300, 2303,65746
 2132 E  RCB  2075
 2427 E  RCC  2223, 2225
 2430 E  RCT  2201, 2227, 2431
 2265 E  RCX  1462, 2007, 2236, 2373
 2266 E  RCY  2010, 2374
 2267 E  RCZ  2011, 2375
```

POST PROCESSOR ASSEMBLY DATA

```
 2431 E  RLC   2226, 2231
77055 E  RST  76643,77007,77045,77051
76750 E  RSU  76646,76675
77121 E CMKO  77075,77077
77155 E CMPS  77144
 2523 E CNAT  77066,77156
 2522 E CNFT  77145,77153
77112 E CNNM  77071
77104 E CNNR  77113,77120
77102 E CNRS  77135
77103 E CNRT  77133,77161
 2524 E CNVA  77142,77155
 2520 E CNXT  77064,77104
   50 E DEMQ   3121, 3122
76647 E EXCA  76630
76640 E EXCB  76635
76636 E EXCC  76650
 2477 E MBIT   2271, 2477
 2340 E MONE   2336,76756
  140 E MONS   2434
 2351 E MOUT   2317, 2333, 2335, 2337
 2111 E MRKA   2104
 2125 E MRKE   2120, 2130
 2505 E MRKP   2122, 2123
 2353 E MRKX   2313
 2316 E MWIN   2334
 2516 E RCAC   2014, 2261, 2371
 2435 E RCBE   2016, 2207, 2212, 2220, 2232, 2257
 2213 E RCEA   2211
 2221 E RCEB   2217
 2236 E RCEC   2233
 2026 E RCIA   2024
 2517 E RCMQ   2015, 2262, 2372
 2413 E RCT1   2243
 2416 E RCT4   2246
 2420 E RCT5   2251
 2424 E RCT6   2254, 2427
 2412 E RCTM   2256
 2410 E RELX   2403
 2142 E SFSA   2152
 2147 E SFSC   2140
 2137 E SFSL   2143
 2301 E SFWA   2154, 2177, 2305
 2175 E SFWB   2276
 2277 E SFWC   2272
 2274 E SFWD   2307
77042 E SUPX 76626,76676,77010,77052
 2433 E TFSC   2221, 2222
 2432 E TFWC   2213, 2214
 2046 E TMLD   2042, 2067
 2050 E TMLE   2045, 2061
 2057 E TMLF   2054
 2061 E TMLG   2052
 2062 E TMLH   2046
 2035 E TMLJ   2063
```

POST PROCESSOR ASSEMBLY DATA

```
 2045 E TMLK   2073
 2503 E TMLM   2033, 2036, 2062, 2064, 2071
 2204 E ZPDL  76670
77143 ECMPLP  77154
77136 ECMPNT  77101
77107 ECNFWX  77060
77110 ECNFWY  77061
77063 ECNMLP  77105
77072 ECNSLP  77103
  176 ECONAT   2507
 3076 EDECON  70240
   47 EDEDIG   3114, 3120, 3125, 3134, 3137, 3143, 3150, 3165
   51 EDEINP   3107, 3131
 3163 EDEIR4   3100, 3103, 3176
   46 EDELOD   3077, 3126, 3135, 3155, 3174
 3201 EDEORG   3201
   44 EDETS1   3076, 3102, 3146
   45 EDETS2   3106, 3127, 3136, 3156, 3172
77056 EEXCIQ  76634
77057 EEXCIR  76637
76626 EEXCIS  65477,76632,76651,77056,77741
76652 EEXCZR  76636,77043
76710 EGETWD  76706
 2101 EMARYA   2106
 2075 EMARYB   2110
 2107 EMARYC   2124, 2131
 2436 EMARYT   2077, 2107
 2436 EMBITF   2477
 2335 EMLBBJ  77006
 2336 EMLBDW  76771
 2334 EMLBFA  77000
 2322 EMLEPD  77003
 2331 EMLEPE   2351,77004
 2354 EMLEXT   2310, 2311
 2333 EMLIST   2315, 2324,76775
 2324 EMLPDC   2027
 2325 EMLPDE   2026
 2350 EMLTBT  76760
  140 EMONES   2172, 2434
 2074 EMPDLF   2065
 2352 EMSRTN   2312
 2674 ENDBLK   2513
 2003 ENDPDL   2023, 3205,14114,76673,77001
11027 ENDRED  11015,65315
15157 ENPDL1  76674
 3523 EQPROG   3522,65363
 2367 ERCBEX   2260
 2355 ERCERR   2325, 2404
 2363 ERCFEM   2361
 2434 ERCIND   2012, 2263, 2367
 2403 ERELOC   2202
10443 EREXIT   1532, 1603,10375,10450
10442 ERNULL   1531,10346
 1670 ERROR1   1071, 1474,65416
10336 ERRSET  65427
```

POST PROCESSOR ASSEMBLY DATA

```
12354 ERSETC   156,10440,65424,65436
76676 ESETUP   115
 2171 ESFWLD  2175,76764
 2271 ESFWSC  2173, 2274
76703 ESLOOP 76717
77022 ESUPFS 77013
77033 ESUPFV 77030
77035 ESUPFW 77032,77037
 2506 ETEMXX  2514
 2504 ETMPTM  2051
10051 EVALCH 10227
10153 EVALQT   104,  116
12330 EVLTSL 14226,65450,65460
10323 EVQEOF 10060
10230 EVQERR 10443
10321 EVQRTS 10051,10103,10234,10236
 1714 EXCABL 65464
 1713 EXCISD  1714,76627,76642
12351 EXP2X2 12300,12311,12344
 1477 EXPLOD 65510
12277 EXP(12 12114,12125
 2203 EZPDLA  2030
   50 F    N  6616, 6620, 6656, 6733, 6735, 6741, 6766, 6775, 7004, 7160, 7162, 7163, 7202
   47 F    T  6612, 6650, 6651, 7035, 7036, 7041, 7042, 7053, 7057, 7060, 7061, 7066, 7077, 7120, 7123, 7125
              7147, 7150,   50
10631 F   B1 10622,10646
10634 F   B3 10625
10637 F   B4 10634
10622 F   B5 10616
10644 F   B6 10636
   44 F   BN  6654, 7000, 7002, 7073, 7210,   45
   46 F   CH  6732, 6737
 6747 F   CM  6730
   44 F   MQ  6624, 6652, 7151,   45
11121 F   T1 10734,10735,11060,11064
11044 F  AL5 11040
10761 F  BB1 10721,10753
10741 F  BB2 10727
10754 F  BB3 10752
10742 F  BB4 10750
10730 F  BB5 10725
 6714 F  BN1  6762
 6654 F  BN2  6717, 7213
 6702 F  BN3  6667, 6720, 7214
   47 F  CHD    50
 7012 F  CM2  6661, 6714, 6715, 6777, 7140
 7014 F  CM3  6664, 6670, 6671, 7012, 7142
 6776 F  CM4  6774
 7015 F  CM5  6776, 7013
 7065 F  CM6  6662, 6716, 7026, 7045
 7050 F  CM7  7027
 7046 F  CM8  7050, 7056, 7064, 7075
 6766 F  CV2  6731, 6750, 6753, 6756, 6761, 6764
 6725 F  CV3  6672, 6676, 6742, 6745
 6744 F  CV4  6743
```

POST PROCESSOR ASSEMBLY DATA

6742	F	CV5	6673, 6675, 6706, 6712, 6713, 6746
6745	F	CV6	6674, 6707
6733	F	CV7	6702, 6722, 7153
6735	F	CV8	6703, 6710
6741	F	CV9	6704
6777	F	CX3	6665, 6770, 7121, 7206, 7207
7005	F	CX5	7003
6632	F	CY2	6644, 6647
6626	F	CY3	6632, 6635
6636	F	CY4	6631
170	F	EOR	11032,11100
6677	F	EX1	6757
6655	F	EX2	6701
6770	F	EXS	6663, 6677, 6700, 7144, 7212
7217	F	FL1	7015, 7022, 7024
7220	F	FL2	7017, 7020, 7301
10712	F	GV1	10647,10666,10675,10706,10720
10670	F	GV2	10662
10714	F	GV3	10661
10710	F	GV4	10700,10702,10703
10677	F	GV6	10671
6721	F	MN1	6751
6724	F	MN2	6723
6711	F	MN3	6724
7220	F	ONE	7033, 7037, 7052, 7055, 7301
6746	F	OVF	6740
6710	F	PL1	6765
6670	F	PT1	6754
6676	F	PT3	6705, 6725
7134	F	PX1	6607, 7047
7135	F	PX2	6610
7136	F	PX4	6611
127	F	QD7	7117, 7215
7140	F	STZ	6767
6750	F	SW1	6660, 7106, 7111, 7211
11062	F	UP1	11056,11057
11053	F	UP2	11061
11073	F	UP3	11063
11063	F	UP4	11072
7132	F	XT1	7112
7116	F	XT2	7113
7112	F	XT3	7107
11114	F	BUFF	10775,11002
7104	F	CM12	7076
7051	F	CM13	7051
7024	F	CMF1	7021
7025	F	CMF2	7023
11161	F	CURC	70240
6737	F	CV10	6711
7215	F	EXC1	7043
7216	F	EXC2	7062
45	F	EXPN	6655, 6772
142	F	FIXS	10710,11130
143	F	FLOS	10664,11121
171	F	HOL9	11037,11100

POST PROCESSOR ASSEMBLY DATA

```
  166 F HORG  10611,10704,11065,11100
 6656 F INTN   6666
11001 F LAMB  10767
11205 F MKIR  11163
11163 F MKNO  70240
 2672 F MKT1  11164,11171,11172,11202
 7155 F OCT1   7164, 7167
 7166 F OCT2   7165
 7202 F OCT3   7172, 7175, 7200
 7204 F OCT5   7176
 7164 F OCT6   7201
 7170 F OCT8   7157
 7162 F OCT9   7154, 7205
10610 F PACK  70240
  131 F PBIT   6721, 6771, 7204, 7215
10772 F PORK  10763,10773,11025,11027
11076 F UPI2  11047
11075 F UPI4  11046
   60 FACTOR  12140,12312,12321,12322,12324,12325,12343,12346
11017 FBACON  11034
10751 FBBIR2  10722,10742
 2671 FBBPNT  10743,10746,10760,11157
10623 FBFLOC  10630,10635,10655,10741,10757
11156 FBOFFO  10624,10656,10744,11120
11162 FCHACT  70240
11130 FCHARS  10613,10617,10627,10645,10651,10723,10755
11005 FCHOPS  10765
11035 FCHPOS  10764,11005,11030
10753 FCLEAR  10667,10676,10707,10713,70240
11160 FCURC1  70240
   51 FDATUM   7032, 7034
 6650 FDECNO   6640, 6643, 6646
11037 FDIGIT  70240
 1260 FENDRD  65605
11131 FEORTS  10770,10776,11033
 7110 FFSTOR   7065, 7143, 7145
  174 FHOL40  11100
 1273 FILDEL  65572
 1102 FILNAM   1160, 1161, 1167, 1170, 1242, 1244, 1245, 1247, 1264, 1265, 5320, 5321,10177
 1252 FILSEK  65633
 7106 FISTOI   7203
 7105 FISTOR   7007, 7011, 7014, 7141, 7146
12775 FIXFLO  11726,11767,12047,12432,12451,12470,12526,12637,12705,12720,12746,12770,13070
11423 FIXVAL   3054,10351,11312,11363,11375,11400,11406,11411,11471,11500,11506
10774 FJOYCE  10771
   27 FLAPCX    107
   75 FLAPCY    111
   76 FLAPCZ     77, 1645
 1651 FLAPDV   1642
 1643 FLAPEX   1653, 1663
 1654 FLAPMQ   1634
 1625 FLAPTR     27
  143 FLOATD   6577,11121,11750,12071,12101,12126,12565,13110
12700 FLOATP  65673
 4523 FLONAM   3752
```

POST PROCESSOR ASSEMBLY DATA

```
13050 FLOTOL 12745
 1662 FLPMQ1  1657
 1650 FLPOUT  1630
 1645 FLPRTN  1652, 1661, 1665
 1667 FLPTMP  1626, 1632, 1643, 1644
   47 FLTBAS 12104,12121,12140,12306,12317,12333
   46 FLTEXP 12070,12073,12112,12122,12140
12131 FLTFLT 12123
10721 FMKNAM 70240
  145 FNARGD 13243,13513
 6607 FNUMBR 10660,70240
10647 FNUMOB 70240
 7206 FOCTIO  7173
 7147 FOCTNO  6641
11120 FPARAM 10657
11117 FPWORD 11003,11007,11011
 7301 FREORG  7301
   52 FRESID  7025, 7040, 7054
  140 FSEVNS 10731,11035
11016 FSHANK 11013,11014
 7076 FSHIFT  7074
15226 FSTATE  1324
11032 FSUZIE 10772
12057 FUNARG   145,65721,65726
 2671 FUPLST 11062,11067,11071,11073,11074,11157
11036 FWDNUM 10766,11001,11031
 3031 FWLOUT  2676
 2713 FWORDL  2153, 2365, 2675, 2701, 2704,77026
 2424 GCPDLC  2047, 2072, 2252, 2427
10010 GENSYM 65770
 1216 GETBCD  1243, 1246, 1275, 1300, 1316, 1321, 1346, 1351,66012
15234 GETBRK  1457
15212 GETCOM  1062,10154,10212,76707
14527 GOGOGO 66023
12660 GRTRTP 12671,66034
 1241 GTFLNM  1253, 1261
 5327 GTGCD1  5401
10456 H     A 10476
10516 H     D 10535
10454 H    AA 10467
10474 H    AD 10503
10514 H    DA 10526
10533 H    DD 10542
11303 H    T1 11210,11212,11214,11220,11224,11226,11230,11234,11240,11242,11244,11250,11266,11271,11274,11275
              11300,11332
11303 H    T2 11307,11320,11332
10466 H   AAX 10505
10502 H   ADX 10507
11275 H   ANS 11235
10461 H   CAX 10450,10463,10470,10477,10504,10506
10520 H   CDX 10510,10522,10527,10536,10543,10545
10525 H   DAX 10544
10541 H   DDX 10546
11276 H   ERS 11251
11274 H   ORS 11221
```

POST PROCESSOR ASSEMBLY DATA

```
10406 H HARP  10417
10440 H HORN  10336,10340,10341,10342,10343,10352,10356,10361,10362,10367,10372,10403,10421,10425,10431,10432
              10434,10436
11255 H LOG1  11265,11301
11252 H LOG2  11222,11236
11272 H LOG4  11253
11264 H LOG5  11252
11266 H LOG6  11302
11330 H LSH1  11305
11316 H LSH2  11314
11324 H LSH3  11316,11317
11327 H LSH4  11306
10367 H OBOE  10357
10417 H TUBA  10404,10405
10403 HBSOON  10374
11207 HLOGOR  70240
    1 HOBTAG  70037,70040,70041,70042,70043,70044,70045,70046,70047,70050,70051,70052,70053,70054,70055,70056
              70060,70061,70062,70063,70064,70065,70066,70067,70070,70071,70072,70073,70074,70075,70076,70077
              70101,70102,70103,70104,70105,70106,70107,70110,70111,70112,70113,70114,70115,70116,70117,70120
              70122,70123,70124,70125,70126,70127,70130,70131,70132,70133,70134,70135,70136
10421 HSHAWM  10402
 6524 I   01  6430, 6431, 6434
 6525 I   02  6436, 6470
 6440 I   03  6444, 6445
 6431 I   04  6441, 6452, 6463, 6467
 6526 I   05  6425, 6504, 6505, 6512
 6452 I   07  6466
 6527 I   Q2  6457, 6465
 6530 I   Q4  6454, 6464
 5047 I   RD  70240
 5233 I  GET  5067, 5134, 5151, 5154, 5166, 5310, 5313
 5264 I  GTX  5233, 5303, 5306
 6472 I  OUT  6432
 5403 I  PUT  5144, 5161, 5165
 5147 I  RDT  5141
 5111 I  RDX  5053
 5116 I  RDY  5054
 5115 I  RDZ  5055
 5024 I  RPI  5004
 5027 I  RP2  5011
 2640 I  RSI  5000, 5022, 5025, 5041
 2641 I  RS2  4773, 4775, 5006, 5015, 5020, 5031, 5037
 5223 I  TPF  5205
 6522 I BSRT  6405, 6426, 6516, 6517
 6404 I BUCK  6423, 6510
 6377 I CELL  1212, 5237, 5254, 5353, 5355, 6547,10214
 5341 I CILA  5331
 5347 I CILB  5343
 5336 I CILD  5342
 5256 I GTMC  5267, 5275
 5266 I GTPC  5255
 5257 I GTPT  5253, 5260, 5261
 6513 I ITRX  6407, 6471
 6514 I ITRY  6410
 5517 I LWPO  5102, 5323, 5332, 5376, 5400
```

POST PROCESSOR ASSEMBLY DATA

```
5371 I MKWD   5346, 5347, 5361, 5365
 151 I OPNA   6477, 6527
5420 I PUTX   5405, 5431
5053 I RDAA   5050
5151 I RDDD   5140
5067 I RDGC   5075
5163 I RDLT   5073
5164 I RDNM   5074
5165 I RDNN   5171, 5172, 5173, 5232
5045 I RDOT   4766, 5007
5225 I RDPD   5177
5201 I RDPS   5175, 5176, 5226, 5230
5110 I RDPU   5076, 5077, 5100, 5217
5203 I RDXT   5160, 5174
4755 I READ   70240
 152 I RLTR   5043
5215 I TPFA   5224
5367 ICIL77   5333, 5334, 5352, 5354
5344 ICILCI   5335
5352 ICILLP   5360
5402 IEOFCT   5325, 5374
5236 IGETGO   5373
5374 IGTEOF   1214, 5324, 5340, 5366
5314 IGTGCD   5235
5277 IGTPC1   5245, 5250
5307 IGTPC2   5300
5516 IGTTBL   5256
6403 IGTVAL   5136, 5143, 5146, 5152, 5156, 5251, 5411
6516 IINTAD   6427
6504 IINTCN   6520
5046 ILRCIS   5125, 5234, 5270, 5304, 5311, 5372, 5452,10217
6406 INTRN1   5222,10640,66047
5422 IPTRFP   5414
5474 IPTSFT   5207, 5413
5432 IPTTFA   5404
5410 IPUTMC   5131, 5203, 5211, 5415, 5423, 5451, 5455,10224
5416 IPUTPC   5212, 5223, 5424, 5430, 5441
5501 IPUTVL   5215, 5216, 5444, 5447
5437 IPUTZB   5407
6400 IRDDDC   5137, 5145, 5153, 5155
5154 IRDDDL   5162
5133 IRDDLR   5072
6401 IRDIND   5056, 5114
5101 IRDJT1   5071, 5200
5200 IRDJT2   5150, 5170
6402 IRDLST   5047, 5051, 5124, 5202, 5460,10216
5501 IRDPNB   5213, 5417, 5425, 5433, 5442, 5502, 5503, 5504, 5505, 5506
5065 IRDPTS   5112, 5123, 5130, 5132, 5454,10223
5046 IRDVAL   5110, 5201
5066 IRDWDS   5113, 5126, 5457,10221
5000 IREAD1   4762, 5014, 5016
4772 IREDER   4765, 4770, 5034, 5036
4760 IREDIS   5030
4757 IREDS1   4755
5043 IRLPAR   4760, 5012
```

POST PROCESSOR ASSEMBLY DATA

```
 5044 IRRPAR  4763, 5002, 5033
 6521 IVALUE  70240
 3513 L  EQA  3464
 3510 L  EQF  3456, 3460, 3474, 3502, 3516
 3505 L  EQT  3454
 3562 L EQL1  3452, 3455, 3470, 3472, 3503, 3513, 3520
 3563 L EQL2  3451, 3453, 3457, 3461, 3465, 3477, 3521
 3453 L EQLP  3504
 3530 L EQPF  3523, 3525
 3561 L EQXR  3450, 3506, 3511, 3517
  146 LABELD 13243
11712 LAMBDA   147,66066,66073
  147 LAMDAD 13243
 1705 LBINPG 76655,76746
 3555 LEQPFX  3535, 3537, 3545, 3547
 3552 LEQPRX  3530, 3555
 3553 LEQPRY  3532, 3556
 3450 LEQUAL 70240
12670 LESSTP 66123
 1710 LFREES 76716,76721
 1707 LFULWS 76725,76737
10201 LISTCM 10166
  100 LISTNW 10172
15204 LITRLS 15204,77747
11223 LOGAND 66145
11237 LOGXOR 66200
64260 LOWERP  1675
 1706 LPBPDL 76666,76744
11305 LSHIFT 66110
 3021 LTBPFJ  3005,77242
 7505 MAPCAR 66222
 7532 MAPCON  7554,66233
 3201 MAPLIS  7452,11442,14136,14234,14376,66244
12725 MINUSP 66312
13051 MNSPRG 66301
 2310 MRKLST  2032, 2056, 2060, 2127,76770,76774
10173 NEWFIL 10162,10164
   31 NILSXX    76
  200 NOBACT  1551, 1610
 1314 NOFILE 65620
 1332 NOFILX  1314, 1335
 1334 NOFILY  1330
10446 NUBPDL  1554,76665
12672 NUMBRP 66401,77200,77472
 3102 NUMNAM  3733
12567 NUMVAL 11261,11321,11702,11710,12063,12075,12417,12632,12673,12702,12715,12726,12736,12762,13052,13062
              77213,77465
64457 OBLIST  2031,64460,66436
10033 OCTALP  2241, 4050
 7542 OLDOBS 64460,64470,64471,64503,64504,64513,64514,64524,64525,64535,64536,64546,64547,64554,64555,64565
              64576,64577,64607,64610,64622,64623,64633,64634,64646,64647,64657,64660,64670,64671,64701,64702
              64713,64723,64724,64734,64735,64745,64746,64756,64757,64767,64770,65000,65001,65011,65012,65022
              65033,65034,65044,65045,65060,65061,65073,65074,65104,65105,65115,65116,65130,65131,65141,65142
              65153,65163,65164,65174,65175,65205,65206,65216,65217,65232,65233,65243,65244,65256,65257,65267
              65300,65301,65311,65312,65324,65325,65335,65336,65346,65347,65357,65360,65370,65371,65401,65402
```

POST PROCESSOR ASSEMBLY DATA

65413,65423,65424,65436,65437,65447,65450,65460,65461,65473,65474,65504,65505,65517,65520,65530
65536,65537,65547,65550,65555,65556,65566,65567,65601,65602,65614,65615,65627,65630,65642,65643
65651,65661,65662,65667,65670,65700,65701,65712,65713,65720,65721,65726,65727,65741,65742,65753
65764,65765,65775,65776,66006,66007,66017,66020,66030,66031,66043,66044,66054,66055,66065,66066
66074,66104,66105,66117,66120,66130,66131,66141,66142,66152,66153,66163,66164,66174,66175,66205
66216,66217,66227,66230,66240,66241,66253,66254,66264,66265,66275,66276,66306,66307,66317,66320
66331,66341,66342,66352,66353,66354,66364,66365,66375,66376,66410,66411,66421,66422,66432,66433
66444,66451,66452,66462,66463,66473,66474,66504,66505,66515,66516,66526,66527,66537,66540,66550
66556,66557,66567,66570,66600,66601,66611,66612,66622,66623,66633,66634,66644,66645,66655,66656
66667,66677,66700,66710,66711,66723,66724,66734,66735,66745,66746,66756,66757,66771,66772,67004
67017,67020,67030,67031,67041,67042,67052,67053,67063,67064,67074,67075,67104,67105,67114,67115
67127,67130,67140,67141,67151,67152,67162,67163,67173,67174,67204,67205,67215,67216,67226,67227
67240,67247,67250,67260,67261,67273,67274,67301,67302,67312,67313,67320,67321,67331,67332,67337
67350,67351,67361,67362,67372,67373,67400,67401,67411,67412,67417,67420,67432,67433,67443,67444
67455,70137,70140,70141,70142,70143,70144,70145,70146,70147,70150,70151,70152,70153,70154,70155
70157,70160,70161,70162,70163,70164,70165,70166,70167,70170,70171,70172,70173,70174,70175,70176
70200,70201,70202,70203,70204,70205,70206,70207,70210,70211,70212,70213,70214,70215,70216,70217
70221,70222,70223,70224,70225,70226,70227,70230,70231,70232,70233,70234,70235,70236,70237

4477 OUTBUF 4423, 4457
1104 OUTFIL 1111, 1112, 1121, 1122, 1130, 1131, 1357, 1360, 1362, 1364, 1372, 1374, 1406, 1407, 1412, 1414
4421,10170,10176
135 PDTMSK 3057,10364,10427
11561 PMAPCA 2612,66217,66227
151 PNAMED 1217, 6527,10032
12112 POSEXP 12107
5276 PRDIND 1206, 1213, 5061, 5327, 5462,10225
2636 PRGVAR 13602,14523
4072 PRINT2 66615
2642 PRINTL 3567, 3572, 4336, 4341
15177 PROSCM 15052,76644,77674,77710,77717
4475 PUNACT 4331, 4434, 4436
1142 PUNCHP 66672
1136 PUNCHT 1143,66661
4412 PUNWRT 4464
1 PWORDS 64466,64477,64511,64522,64533,64544,64552,64563,64574,64605,64616,64631,64642,64655,64666,64677
64721,64732,64743,64754,64765,64776,65007,65020,65031,65042,65054,65067,65102,65113,65124,65137
65161,65172,65203,65214,65224,65241,65252,65265,65276,65307,65320,65333,65344,65355,65366,65377
65421,65432,65445,65456,65467,65502,65513,65526,65534,65545,65553,65564,65575,65610,65623,65636
65657,65665,65676,65710,65716,65724,65735,65751,65762,65773,66004,66015,66026,66037,66052,66063
66102,66113,66126,66137,66150,66161,66172,66203,66214,66225,66236,66247,66262,66273,66304,66315
66337,66350,66362,66373,66404,66417,66430,66441,66447,66460,66471,66502,66513,66524,66535,66546
66565,66576,66607,66620,66631,66642,66653,66664,66675,66706,66717,66732,66743,66754,66765,67000
67026,67037,67050,67061,67072,67102,67112,67122,67136,67147,67160,67171,67202,67213,67224,67235
67256,67267,67277,67310,67316,67327,67335,67346,67357,67370,67376,67407,67426,67441,67452
12215 Q A 12162
12216 Q B 12164
12217 Q C 12170
12220 Q D 12173
51 Q F 12140,12157,12160,12174,12176
50 Q M 12140,12155,12204
52 Q T 12140,12163,12165,12166,12172,12175,12200
12271 Q AA 12245
11416 Q AX 11343
12272 Q BB 12247
12275 Q CU 12223,12226

POST PROCESSOR ASSEMBLY DATA

```
12273 Q    DD 12253
13005 Q    FL 13000
13011 Q    FX 12776
12265 Q    K2 12227
12266 Q    K3 12234
12267 Q    K4 12236
12270 Q    K5 12257,12260
   53 Q    SQ 12140,12161,12167
12204 Q    T1 12144
12650 Q   A1T 12633,12636,12643
11606 Q   AAA 11611
12630 Q   AD1 12657
11466 Q   ADA 11463,11464
11670 Q   ADD 11472,11512,11516,11522,11525,11527,11540,11565,11576
11671 Q   ADT 11501,11515,11523,11531,11577
12543 Q   AFL 12530
11527 Q   AGA 11511,11520
11357 Q   AGN 11345,11346
11413 Q   AGR 11401,11404
11417 Q   AGV 11347,11366
12525 Q   AMM 12430
 2663 Q   AMQ 12420,12431,12450,12467,12525
12565 Q   AMR 12516,12522
11512 Q   AOD 11474
11677 Q   ARY 11620
  153 Q   ASB 11630,11700
11521 Q   ATD 11503
11415 Q   AXS 11356
12214 Q   CHI 12202
12140 Q   EXP 12136,65523
12651 Q   FAD 12641
12711 Q   FLT 12707
12221 Q   LOG 12133,66167
12375 Q   MAX 70240
12367 Q   MIN 70240
12675 Q   NPT 12731
12153 Q   SHI 12151
12344 Q   TXL 12331
13046 Q   UFC 13012,13015,13035,13040
13023 Q   UFE 13014
13031 Q   UFF 13026,13033,13034
13045 Q   UFQ 13017,13021
13047 Q   UFS 13025,13030,13041
12751 Q   ZPF 12710,12722,12757
12741 Q   ZPG 12774
12754 Q   ZPT 12742,12756
11577 Q  AADD 11602
11674 Q  AARY 11562,11612
11570 Q  ACLA 11573
12626 Q  ADD1 70240
12653 Q  ADDF 12642
12353 Q  ADDP 70240
11665 Q  ADDT 11532,11537,11570,11603
11672 Q  ADTH 11507,11513,11521,11533,11606
 2664 Q  AFAT 11453,11633,11641,11647,11654
```

POST PROCESSOR ASSEMBLY DATA

```
12460 Q AFLL 12434
12462 Q AFLR 12436
12507 Q AFLT 12474
12505 Q AFMP 12472
11650 Q AFRX 11444
11651 Q AFRY 11445,11653
11420 Q AGAO 11334,11344,11351,11362,11377,11410
11421 Q AGAT 11340,11353,11402,11405
11411 Q AGD1 11403
11404 Q AGDT 11361
11366 Q AGXE 11414
12547 Q AMIN 12535
 2660 Q AMIR 12353,12361,12367,12375,12402,12562
12411 Q AMLP 12444,12457,12504,12542,12554
12465 Q AMLT 12424
12442 Q AMRT 12464
12502 Q AMRU 12512
11662 Q APWO 11552,11554
11663 Q APWT 11542,11545
11371 Q ARYY 11332
11372 Q ARYZ 11333
11673 Q ASBR 11555,11557,11623
11664 Q ATBZ 11553,11566
 2665 Q ATMP 11470,11473,11476,11502,11550,11622,11625
11667 Q ATMQ 11543,11556
11666 Q ATYP 11462,11465,11534,11535,11544
12213 Q CHAR 12145,12150
12313 Q CLA1 12337,12342
12326 Q CLA2 12315,12323
12057 Q DIFT 12046,12051,12052
12055 Q DIFX 12042
12026 Q DIVA 11776
12041 Q DIVT 11766,11771,11777,12007,12010,12013,12024
12031 Q DIVX 11762,12015,12035
12060 Q EXPT 70240
12713 Q FIXP 70240
11434 Q FXVE 11423,11426,11431
12666 Q GRTT 12662
12210 Q HUGE 12206
12276 Q LOG2 12263
12212 Q LOGE 12152
12211 Q MAXX 12141
13056 Q MRXR 13051
12361 Q MULT 70240
12523 Q MXIR 12515
12676 Q NPIR 12672,12674,12700,12713,12725,12733,12734,12753,12760
12576 Q NVNO 12572,12611
12760 Q ONEP 70240
  143 Q RCPS 13100,13110
13107 Q RCPT 13065,13074
13101 Q RRXR 13060,13106
12655 Q SUB1 70240
  120 Q UFMC 13016,13020,13045
  117 Q UFNC 13032,13045
13043 Q UFXR 13023
```

POST PROCESSOR ASSEMBLY DATA

```
12513 Q UNFX 12440,12476,12520,12533
11716 Q UNUE 11744
11747 Q UNUR 11706,11714,11716,11725
11745 Q UNUS 11705,11720,11731
11746 Q UNUT 11701,11707,11723,11735
11721 Q UNUX 11700,11737
12274 Q VLMP 12243
12752 Q ZPIR 12701,12712,12714,12724,12735,12755,12761
12756 Q ZPTS 12747,12750
  120 Q233Q9 13045
  121 Q777Q9 13045
12644 QA1IR1 12626,12655
12645 QA1IR2 12630
12646 QA1IR4 12631
11675 QACLAS 11575
11422 QAGATH 11342,11355,11374
11374 QAGDTH 11360
11367 QAGXEX 11336,11357
12555 QAMEND 12412
11443 QAMFAG 11441
  143 QAMFLC 12560,12565
12445 QAMFRS 12422
  142 QAMFXC 12556,12565
 2661 QAMIND 12355,12363,12371,12377,12404,12561
12563 QAMIR2 12407
 2662 QAMLIS 12414,12443,12456,12503,12541,12553
12402 QAMMMF 12360,12366,12374
12534 QAMRNT 12546
12566 QAMSUM 12410,12441,12442,12447,12463,12500,12502,12511,12517,12521,12536,12537,12547,12552,12555
11676 QARSTU 11605
11653 QARYTL 11547
   54 QCMMON 12140,12230,12235,12237,12240,12241,12242,12244,12246,12250,12252,12256,12261
12054 QDIFX2 12043
11773 QDIVDC 12025
12034 QDIVEX 12003
12021 QDIVFX 11772
12040 QDIVND 11750,11753,11757,12017,12032,12036
11762 QDIVOP 11752,11756
12030 QDIVX2 11763,12016,12034
12126 QEXPFL 12131,12137
   45 QEXPNT 12066,12106,12124,12135,12140,12310,12313,12326,12330
12116 QEXPTX 12060,12127
12130 QFXFLT 12113
12340 QFXMP2 12334
12324 QFXMPY 12320
12664 QGRTIR 12660,12667
12206 QLARGE 12142
12515 QMIXFL 12462,12507,12545
12607 QNVATM 12575
12617 QNVIR4 12567,12601
12620 QNVTBL  1151,12616
   57 QPOWER 12140,12304,12316,12332,12335,12336,12340,12341
   44 QPTRBS 12061,12074,12140
13104 QRCPFX 13073
13102 QRRXR2 13061
```

POST PROCESSOR ASSEMBLY DATA

```
  154 QSPECD 77606,77626
15173 QTLSTC 76645,77544,77570,77576
  121 QUFMSK 13013,13045
13012 QUNFIX 70240
  143 QUNFLT 11733,11750
11723 QUNMXA 11715
11740 QUNMXB 11730
11736 QUNUX2 11724,11743
  152 QUOTED  5043
11757 QUOTEN 66714
12734 QZEROP 70240
 3346 R   C1  3342
 2657 R   ST  7354, 7360, 7363
 2655 R   SX  7350, 7424
 2656 R   SY  7351, 7355
 2654 R   SZ  7366, 7367, 7373, 7377, 7401, 7404, 7405, 7407, 7416
 7767 R  AT1  7755
 3323 R  CMP  3230
 2567 R  CS1  3335, 3344, 3364
 2570 R  CS2  3347, 3352, 3355, 3356, 3361
 3366 R  CT1  3337, 3343, 3350
13674 R  LQP 70240
14547 R  GOT 14536
 2604 R  GOX 14527,14542,14550
 2621 R  MS1  3202, 3211, 3267, 3272
 2622 R  MS2  3213, 3226, 3250, 3265, 3276
 2623 R  MS3  3215, 3227, 3231, 3263, 3277, 3301
 2624 R  MS4  3217, 3241, 3254, 3261
 2625 R  MS5  3221, 3242, 3257, 3317, 3321, 3333
 3333 R  MS6  3223
 7752 R  NLY  7721
 3424 R  SR1  3410
 3373 R  SR3  3433
 3434 R  SR4  3373
 2653 R  SXT  7353, 7422
 7751 R  TWA  7726
13470 R ATMX 13457
13465 R ATP1 13460
 7761 R ATRB  7756, 7764
  133 R BFDM 13454,13457
  126 R BFQ1 13465,13472,13475,13677
13450 R BFS1 13477,13511,13513,13536,13575,13607,13674,13675
 2606 R BFS2 13500,13505,13517,13526,13577,13605
 2607 R BFS3 13514,13532
 2605 R BFS4 13524,13535
13616 R BFS5 13576,13612
13441 R CARP  1221,70240
13446 R CARX 13441
13451 R CDRP 70240
13455 R CDRX 13451
 3327 R CMP1  3300
 3334 R COPY  3354, 3360,70240
  133 R DECM  3353, 3367, 3432
 2553 R EVA1 13622,13637
 2554 R EVA2 13627,13641
```

POST PROCESSOR ASSEMBLY DATA

```
13641 R EVA3 13635
13626 R EVA4 13642
13636 R EVA5 13644
13622 R EVA6 13617
13617 R EVA8 70240
 2555 R EVA9 13624,13632,13634
  126 R EVCT 13620,13643,13664,13674
 2626 R EVR1 13650,13666
 2627 R EVR2 13655,13670
13670 R EVR3 13663
13654 R EVR4 13671
13665 R EVR5 13673
13650 R EVR6 13645
13645 R EVR8 70240
 2630 R EVR9 13652,13660,13662
 2634 R INTB 14371,14420,14423,14441,14447,14470
14522 R INTE 14372,14403,14406
13513 R LABP 70240
13477 R LAMP 70240
 3232 R MAIN  3326
  133 R MCDM  7552, 7562
 2620 R MCN2  7535, 7547, 7556
 2617 R MCN3  7536, 7550
 2616 R MCN4  7537, 7540, 7542, 7553
 2615 R MCN5  7533, 7560
  133 R NCDM  7571, 7600
 7565 R NCI1  7562
 7567 R NCI2  7572
 7576 R NCS1  7565
 7600 R NCS3  7566, 7575
 7632 R PROP 70240
 2651 R REPV 13542,13546,13560,13571
13575 R SCTP 70240
 7734 R SPP1  7742
 7753 R SPRX  7714
 7747 R SPRY  7730
 7746 R SPRZ  7731, 7734
 2643 R SRS1  3367, 3422, 3437, 3446
 2644 R SRS2  3374, 3411, 3430
 2645 R SRS3  3372, 3401, 3403
 2646 R SRS4  3376, 3412, 3414, 3426
 2647 R SRS5  3371, 3400, 3424
 7353 R SUB1  7376, 7403
 7422 R SUB2  7364, 7425
 7420 R SUB3  7410
 7424 R SUB4  7357
13457 RATOMP 70240
  145 RBFFAG 13510,13513
 2261 RCEXIT  2235
13060 RCPPRG 66751
 2430 RCRLOC  2431, 3033, 3040
 2500 RCSGNL  2017
 2501 RCSGNM  2134
 2502 RCSGNN  1451, 2013, 2200, 2264, 2370
 1211 RDFLX1  5315
```

POST PROCESSOR ASSEMBLY DATA

```
15224 RDFLXA  1211,10774
 1207 RDFLXX  1202
 1215 RDINDC  1203, 1205, 5063, 5121, 5272, 5314, 5464,10226
  133 RDMASK  7763, 7771
 5124 RDMOS1  5064, 5122
 5120 RDMOSE  5062
15240 RDWAIT  5317
 2007 RECLAM  3034, 3041,15021,66762
 1067 REDOIO  1065, 1107, 1116, 1126, 1156, 1164, 1262, 1302, 1323, 1355, 1404, 4416, 5316
11753 REMAIN 66775
 7601 REMPRP 67010
15163 RESTOR 15202
14523 RETURN 67023
14417 RINTAA 14404
14433 RINTEB 14446
14361 RINTER 70240
14452 RINTEV 14430,14431
14473 RINTFB 14375
14504 RINTFC 14462
14502 RINTFX 14474
14420 RINTGA 14427,14433,14456,14472
 2635 RINTGL 14364,14417,14464,14507,14516
14403 RINTGM 14412,14416
 2637 RINTGS 14363,14366,14455,14457,14471,14505,14513,14515,14524,14532,14537,14546
 2636 RINTPL 14365,14377,14401,14444,14453,14523
14513 RINTRN 14421,14460
 2633 RINTRX 14361,14520
 3302 RMAIN1  3332
 3250 RMLOP1  3322
 3275 RMPRG1  3253
 7562 RNCONC 70240
13472 RNULLP 70240
 7771 RPLACA 67045
10000 RPLACD 67056
10004 RPLACW 67067
13570 RREPP1 13555
 2650 RREPS1 13540,13566
 2652 RREPT1 13554,13563
 7611 RRMPR1  7621
 7613 RRMPR2  7610, 7630
 7624 RRMPRE  7617
 7616 RRMPRT  7604, 7606
 7631 RRMPRX  7601, 7622
  133 RSASDM  7646, 7713
 7640 RSASL1  7644, 7645
 7645 RSASP1  7634
 7644 RSASP2  7636
 7651 RSASP3  7641, 7712
 7710 RSASP4  7673
 7673 RSASP5  7701, 7702
 7701 RSASP6  7670
 7702 RSASP7  7666
  126 RSASQ1  7635, 7667, 7713
 7706 RSAST1  7632, 7647, 7654, 7661, 7663
 7705 RSAST2  7664, 7710
```

POST PROCESSOR ASSEMBLY DATA

```
 7703 RSAST3  7665, 7711
13611 RSETP1 13600
13540 RSETQP 70240
 7350 RSUBST 70240
10205 S COMM 10175
10070 S EVQA 10064
 2670 S EVQB 10052,10063,10065,10100
10131 S EVQD 10117,10315
10104 S EVQS 10143
10130 S EVQZ 10267,10303
  133 S GENC 10026,10032
10030 S GENX 10010
 7776 S REPL  7771,10000,10004
 7663 SASSOC 13325,13561,13603,14465,70240,77462
15230 SAVBRK  1437
10206 SCOMLP 10215
10032 SDIGIT 10011,10013
 3367 SEARCH 67155
 1174 SEEKRT  1155
15232 SETBRK  1444, 1470
15204 SETMEM   114
76653 SETUP1 76747
10050 SEVALQ 10200
  140 SEVENS  2434, 3166, 4020, 4111, 4755, 5206,10161,10644,10652,11035,11233
10316 SEVQAC 10122,10127
 2666 SEVQAN 10133,10136,10140,10241,10270,10302
10244 SEVQAT 10126,10260,10261
10145 SEVQDN   101, 2366,10142
10141 SEVQER 10235
10144 SEVQFN 10066,10070
10304 SEVQFS 10252,10257
10132 SEVQFT 10116,10121,10310
10262 SEVQFX 10254
 2667 SEVQLS 10053,10075,10077,10104,10106,10141
10317 SEVQMQ 10271,10277
10127 SEVQNF 10245
10102 SEVQPU 10057,10243
10054 SEVQRD 10067,10101
10324 SEVQRE 10240
10057 SEVQRF 10062
10322 SEVQSP 10055
10320 SEVQST 10250,10255,10266,10313
10167 SEVQT1 10160
  151 SGENPN 10023,10032
10213 SGETWD 10211
10540 SPECAL   154,65230,67240,67247,77604,77605
 7713 SPREAD 10314,13306,14202
 3441 SRCMPT  3436
 7775 SRPLEX 10003,10007
10773 STREAD 67264
 2153 SWPFWS  2146
 4061 T    L  3741, 3745
 4742 T   A1  4557, 4603, 4605, 4610, 4611, 4615, 4620, 4624, 4632, 4650
 3644 T   A2  3633
 3622 T   A3  3650
```

POST PROCESSOR ASSEMBLY DATA

3627	T	A4	3653
3640	T	A6	3630
4754	T	C0	4533, 4574
4745	T	C7	4572
3664	T	L1	3602, 3607
6545	T	NE	6557
4314	T	ADT	4314, 4315, 4316, 4317, 4320, 4321, 4322, 4323, 4324, 4325
4064	T	BQ0	4036
3654	T	DOT	3634
6536	T	NA1	6544
6553	T	NA2	6543
6566	T	NA3	6562
6577	T	NA7	6560
6601	T	NA8	6565, 6570, 6572, 6573
6602	T	NX1	6533, 6576
6603	T	NX2	6532
6604	T	NX4	6531
3735	T	PA3	3712
4353	T	PLP	4373
4377	T	PNX	4344, 4437
4400	T	PNY	4345, 4440
4401	T	PNZ	4346, 4441
4060	T	PR1	3665, 3747, 4055
3705	T	PR3	3677
3740	T	PR4	3746
4045	T	PR5	3714
4270	T	PR7	4103, 4145
4271	T	PR8	4102, 4146, 4230, 4254
4272	T	PR9	4072, 4101, 4147, 4231, 4255
3656	T	PS1	3577, 3610, 3642
3657	T	PS2	3621, 3622, 3627, 3636, 3646
4313	T	REC	4152, 4160, 4215, 4240, 4247
4521	T	S57	4447, 4522
4167	T	TAB	4170
3612	T	XA1	3606
132	T	ADDM	3675, 3676, 3705
4067	T	BCIQ	4025
4063	T	BQ10	4031
4136	T	COMB	4142
3663	T	CWRL	3604, 3617
4003	T	FIFI	3777, 4014, 4016
4660	T	FL01	4653
4646	T	FL65	4660
4674	T	FL70	4672
4565	T	FL75	4556
4573	T	FL76	4576
4607	T	FL77	4602
4615	T	FL79	4613
4621	T	FL80	4614
4642	T	FL81	4636, 4637
4644	T	FL82	4626
4535	T	FLNA	4526
4711	T	FLNX	4523, 4527, 4705
4712	T	FLNY	4535
4713	T	FLNZ	4536

POST PROCESSOR ASSEMBLY DATA

```
4262 T INIT   4105
4043 T JUDY   3754
4123 T JUST   4116
4743 T LOG2   4554
3754 T LUCY   3731
 140 T ONES   4677, 4755
4275 T PART   4134, 4137, 4164, 4210, 4234, 4257, 4264
4473 T PCNT   4370, 4452
4373 T PGRA   4405, 4407, 4433
4474 T PNCQ   4362, 4410, 4432
4521 T POUP   4371, 4426, 4453, 4522
4374 T POUT   4367
3720 T PR3N   3706
3711 T PR3P   3717
3747 T PR4E   3734, 3753, 4044
3751 T PR4F   3726
3651 T PRP2   3624
4057 T PRSS   3666, 3672, 3720, 4045
4344 T PUN2  70240
4365 T PUND   4357, 4361
4220 T TABA   4203
4215 T TABB   4220
4226 T TABC   4177, 4205
4222 T TABX   4167, 4176, 4227
4223 T TABY   4170
4276 T TEMP   4107, 4110, 4117, 4121, 4123, 4132, 4133, 4156, 4161
4236 T TER1   4237
4244 T TER2   4241
4240 T TER3   4243
4062 T TONI   3775, 4000, 4006, 4011, 4015
4450 T TPLP   4363, 4454
4167 TABTAB   4170,67343
 137 TAGMSK   3536, 3546,12610,14561,14562,77070
4065 TBCIOQ   4041
3764 TBETTY   3757
4106 TCOMB4   4267
4141 TCOMB5   4155, 4166
3662 TCOMM2   3644
10447 TCOUNT  2736, 2747, 3044, 3066,10373
4020 TDEBBY   4001
2514 TEMLIS   2034
   1 TEMTEM   2525, 2526, 2527, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2540, 2541, 2542, 2543, 2544
              2546, 2547, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2560, 2561, 2562, 2563, 2564, 2565
              2567, 2570, 2571, 2572, 2573, 2574, 2575, 2605, 2606, 2607, 2612, 2613, 2614, 2615, 2616, 2617
              2626, 2627, 2630, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2673
5452 TEREAD   1163, 1267, 1472, 4774, 5103,10102,10171,10230
1777 TERPDL  10233
4230 TERPRI   1606, 3571, 3614, 3671, 3701, 4226,10231,14305,14340,67354
4434 TERPUN   4340,10232
4600 TFL76A   4567
4731 TFLOPB   4541, 4542, 4543, 4700, 4706, 4720
4725 TFLZET   4645, 4652, 4654, 4662
4731 TFLZPZ   4531
3775 TGRETA   4005
4716 TINBCD   4627, 4634, 4641, 4643, 4656, 4663, 4665, 4667, 4674
```

POST PROCESSOR ASSEMBLY DATA

```
 6606 TKBPOS   6555
 4744 TLOG10   4565
 3661 TLPAR2   3615
 4132 TLSHIF   4124
 4041 TMARIE   3765
 4027 TMICKY   4024
 4066 TMISGN   3760
 4145 TNOJOB   4113
 6531 TNUTRN  70240
 4274 TPARTS   4135, 4143, 4206, 4214, 4232, 4253, 4266
 4042 TPATSY   4026, 4033, 4040
 4313 TPCPPI   3566
 4342 TPNCHX   4327
 4350 TPPRTS   4374, 4445
 3655 TPRBLW   3575, 3651
 3577 TPRINO   3570, 3626,70240
 3665 TPRIN1  70240
 4076 TPRIN2  70240
 3564 TPRINT  70240
 3575 TPRNIL   3600
 4403 TPRPLP   4372
 3573 TPRPS1   3564
 4070 TPTPNT   3723, 3755
 4071 TPTTGR   3674, 3705, 3724, 3727
 4327 TPUNCH  70240
 4522 TPUNLM   4330, 4355, 4406, 4431
 4455 TPUNWD   4364, 4411
 4347 TPWRDS   4375, 4444
 1116 TRAOPN   1113, 1114
 4156 TRECFL   4153
 3660 TRPAR2   3641
 4034 TSANDY   4027
 4114 TSHIFL   4122
 4024 TVICKI   4017
 4152 TWFULL   4140
 4273 TWORDS   3612, 3667, 4104, 4144, 4204, 4221, 4233, 4251
 3066 UNCONT  67423
11046 UNPACK   1513,67436
 1736 UNSAVE   1565, 3363, 3421, 3434, 3501, 3640, 5021, 5040, 7421, 7444, 7527, 7557,10413,10435,11216,11232
               12406,13144,13316,13370,13432,13565,13636,13665,14060,14204,14235,14254,14273,14517,14544,14653
11700 UNUMIX  11764,12044,12661
70240 UPERML   1673, 1674,70240,76651
 2477 VERBOS   1176, 2234
 4416 WRITIT   4413, 4462
 1144 WRMODE   1123, 1137, 1171
15236 WRWAIT   4417
 1720 X EXER   1715
  102 XLDABS     26
77740 ZEROXC  77044
```

NO ERROR IN ABOVE ASSEMBLY.