fig-FORTH FOR 6800

ASSEMBLY SOURCE LISTING

RELEASE 1

MAY 1979

WITH COMPILER SECURITY

AND VARIABLE LENGTH NAMES

This public domain publication is provided
through the courtesy of:

P.O. Box 8231 • San Jose, CA 95155 • (408) 277-0668

```
                        NAM    Copyright:FORTH Interest Group
                        OPT    NOG,PAG
                      * filename FTH7.21
                      * === FORTH-6800  06-06-79  21:00


                      *
                      * This listing is in the PUBLIC DOMAIN and
                      * may be freely copied or published with the
                      * restriction that a credit line is printed
                      * with the material, crediting the
                      * authors and the FORTH INTEREST GROUP.
                      *
                      * === by Dave Lion,
                      * ===  with help from
                      * === Bob Smith,
                      * === LaFarr Stuart,
                      * === The Forth Interest Group
                      * === PO Box 1105
                      * === San Carlos, CA 94070
                      * ===  and
                      * === Unbounded Computing
                      * === 1134-K Aster Ave.
                      * === Sunnyvale, CA 94086
                      *
                      *  This version was developed on an AMI EVK 300 PROTO
                      *  system using an ACIA for the I/O.  All terminal I/O
                      *  is done in three subroutines:
                      *   PEMIT  ( word # 182 )
                      *   PKEY   (        183 )
                      *   PQTERM (        184 )
                      *
                      *  The FORTH words for disc related I/O follow the model
                      *  of the FORTH Interest Group, but have not been
                      *  tested using a real disc.
                      *
                      *  Addresses in this implementation reflect the fact that,
                      *  on the development system, it was convenient to
                      *  write-protect memory at hex 1000, and leave the first
                      *  4K bytes write-enabled.  As a consequence, code from
                      *  location $1000 to lable ZZZZ could be put in ROM.
                      *  Minor deviations from the model were made in the
                      *  initialization and words ?STACK and FORGET
                      *  in order to do this.
                      *


                      *
0004                  NBLK   EQU    4          # of disc buffer blocks for virtual memory
3210                  MEMEND EQU    132*NBLK+$3000 end of ram
                      *  each block is 132 bytes in size,
                      *  holding 128 characters
                      *
3FFF                  MEMTOP EQU    $3FFF      absolute end of all ram
FBCE                  ACIAC  EQU    $FBCE      the ACIA control address and
```

```
FBCF            ACIAD  EQU    ACIAC+1    data address for PROTO
                *
```

```
*         MEMORY MAP for this 16K system:
*  ( positioned so that systems with 4k byte write-
*    protected segments can write protect FORTH )
*
* addr.              contents                      pointer  init by
* ****    ****************************            *******  ******
* 3FFF                                                      HI
*         substitute for disc mass memory
* 3210                                                      LO,MEMEND
* 320F
*         4 buffer sectors of VIRTUAL MEMORY
* 3000                                                      FIRST
* >>>>>>  memory from here up must be RAM  <<<<<<
*
* 27FF
*         6k of romable "FORTH"                <== IP       ABORT
*                                              <== W
*         the VIRTUAL FORTH MACHINE
*
* 1004  <<< WARM START ENTRY >>>
* 1000  <<< COLD START ENTRY >>>
*
* >>>>>>  memory from here down must be RAM  <<<<<<
* FFE     RETURN STACK base                    <== RP       RINIT
*
* FB4
*         INPUT LINE BUFFER
*         holds up to 132 characters
*         and is scanned upward by IN
*         starting at TIB
* F30                                          <== IN       TIB
* F2F     DATA STACK                           <== SP       SP0,SINIT
*   |     grows downward from F2F
*   |
*   V
*
*   ^
*   |
*   |     DICTIONARY grows upward
*
* 183     end of ram-dictionary.               <== DP       DPINIT
*         "TASK"
*
* 150     "FORTH" ( a word )            <=,    <== CONTEXT
*                                          `==== CURRENT
* 148     start of ram-dictionary.
*
* 100     user #1 table of variables           <== UP       UPINIT
*  F0     registers & pointers for the virtual machine
*         scratch area used by various words
*  E0     lowest address used by FORTH
*
* 0000
```

```
                * * *
                *
                * CONVENTIONS USED IN THIS PROGRAM ARE AS FOLLOWS:
                *
                * IP   points to the current instruction  ( pre-increment mode )
                * RP   points to second free byte (first free word) in return stack
                * SP   (hardware SP) points to first free byte in data stack
                *
                *      when A and B hold one 16 bit FORTH data word,
                *      A contains the high byte, B, the low byte.
                * * *



00E0                    ORG    $E0        variables


00E0            N       RMB    10         used as scratch by (FIND),ENCLOSE,CMOVE,EMIT,KEY,
                *                                 SP@,SWAP,DOES>,COLD


                *         These locations are used by the TRACE routine :


00EA            TRLIM   RMB    1          the count for tracing without user intervention
00EB            TRACEM  RMB    1          non-zero = trace mode
00EC            BRKPT   RMB    2          the breakpoint address at which
                *       the program will go into trace mode
00EE            VECT    RMB    2          vector to machine code
                *       (only needed if the TRACE routine is resident)


                *       Registers used by the FORTH virtual machine:
                *       Starting at $00F0 :


00F0            W       RMB    2          the instruction register points to 6800 code
00F2            IP      RMB    2          the instruction pointer points to pointer to 6800 code
00F4            RP      RMB    2          the return stack pointer
00F6            UP      RMB    2          the pointer to base of current user's 'USER' table
                *             ( altered during multi-tasking )
                *
```

```
                *       This system is shown with one user, but additional users
                *       may be added by allocating additional user tables:
                *       UORIG2   RMB 64  data table for user #2
                *
                *
                *       Some of this stuff gets initialized during
                *       COLD start  and  WARM start:
                *       [ names correspond to FORTH words of similar (no X) name ]
                *
0100                    ORG     $100
0100            UORIG   RMB     6           3 reserved variables
0106            XSPZER  RMB     2           initial top of data stack for this user
0108            XRZERO  RMB     2           initial top of return stack
010A            XTIB    RMB     2           start of terminal input buffer
010C            XWIDTH  RMB     2           name field width
010E            XWARN   RMB     2           warning message mode (0 = no disc)
0110            XFENCE  RMB     2           fence for FORGET
0112            XDP     RMB     2           dictionary pointer
0114            XVOCL   RMB     2           vocabulary linking
0116            XBLK    RMB     2           disc block being accessed
0118            XIN     RMB     2           scan pointer into the block
011A            XOUT    RMB     2           cursor position
011C            XSCR    RMB     2           disc screen being accessed ( 0=terminal )
011E            XOFSET  RMB     2           disc sector offset for multi-disc
0120            XCONT   RMB     2           last word in primary search vocabulary
0122            XCURR   RMB     2           last word in extensible vocabulary
0124            XSTATE  RMB     2           flag for 'interpret' or 'compile' modes
0126            XBASE   RMB     2           number base for I/O numeric conversion
0128            XDPL    RMB     2           decimal point place
012A            XFLD    RMB     2
012C            XCSP    RMB     2           current stack position, for compile checks
012E            XRNUM   RMB     2
0130            XHLD    RMB     2
0132            XDELAY  RMB     2           carriage return delay count
0134            XCOLUM  RMB     2           carriage width
0136            IOSTAT  RMB     2           last acia status from write/read
0138                    RMB     2           ( 4 spares! )
013A                    RMB     2
013C                    RMB     2
013E                    RMB     2



                *
                *
                *    end of user table, start of common system variables
                *
                *
                *
0140            XUSE    RMB     2
0142            XPREV   RMB     2
0144                    RMB     4           ( spares )
                *
```

```
                * These things, up through the lable 'REND', are overwritten
                * at time of cold load and should have the same contents
                * as shown here:
                *
0148 C5               FCB   $C5         immediate
0149 46               FCC   4,FORTH
014D C8               FCB   $C8
014E 27 40            FDB   NOOP-7
0150 19 F5      FORTH FDB   DODOES,DOVOC,$81A0,TASK-7
0158 00 00            FDB   0
                *
015A 28               FCC   "(C) Forth Interest Group, 1979"

0178 84               FCB   $84
0179 54               FCC   3,TASK
017C CB               FCB   $CB
017D 01 48            FDB   FORTH-8
017F 15 25      TASK  FDB   DOCOL,SEMIS
                *
0183            REND  EQU   *          ( first empty location in dictionary )
```

```
                *    The FORTH program ( address $1000 to $27FF ) is written
                *    so that it can be in a ROM, or write-protected if desired
1000                  ORG    $1000

                * ######>> screen 3 <<
                *
                ****************************
                **  C O L D   E N T R Y   **
                ****************************
1000 01         ORIG   NOP
1001 7E 1F 96          JMP    CENT

                ****************************
                **  W A R M   E N T R Y   **
                ****************************
1004 01                NOP
1005 7E 1F C6          JMP    WENT       warm-start code, keeps current dictionary intact


                *
                ****** startup parameters **********************
                *
1008 68 00             FDB    $6800,0000 cpu & revision
100C 00 00             FDB    0          topmost word in FORTH vocabulary
100E 00 7F      BACKSP FDB    $7F        backspace character for editing
1010 01 00      UPINIT FDB    UORIG      initial user area
1012 0F 30      SINIT  FDB    ORIG-$D0   initial top of data stack
1014 0F FE      RINIT  FDB    ORIG-2     initial top of return stack
1016 0F 30             FDB    ORIG-$D0   terminal input buffer
1018 00 1F             FDB    31         initial name field width
101A 00 00             FDB    0          initial warning mode (0 = no disc)
101C 01 83      FENCIN FDB    REND       initial fence
101E 01 83      DPINIT FDB    REND       cold start value for DP
1020 01 58      VOCINT FDB    FORTH+8    cold start value for VOC-LINK
1022 00 84      COLINT FDB    132        initial terminal carriage width
1024 00 04      DELINT FDB    4          initial carriage return delay
                ************************************************
                *
```

```
                *
                * ######>> screen 13 <<
1026 32         PULABX PUL A              24 cycles until 'NEXT'
1027 33                PUL B
1028 A7 00      STABX  STA A  0,X         16 cycles until 'NEXT'
102A E7 01             STA B  1,X
102C 20 06             BRA    NEXT
102E A6 00      GETX   LDA A  0,X         18 cycles until 'NEXT'
1030 E6 01             LDA B  1,X
1032 37         PUSHBA PSH B              8 cycles until 'NEXT'
1033 36                PSH A



                *
                * "NEXT" takes 38 cycles if TRACE is removed,
                *
                * and 95 cycles if NOT tracing.
                *
                * = = = = = = =  t h e  v i r t u a l  m a c h i n e  = = = = =
                *                                                             =
1034 DE F2      NEXT   LDX    IP
1036 08                INX              pre-increment mode
1037 08                INX
1038 DF F2             STX    IP
103A EE 00      NEXT2  LDX    0,X        get W which points to CFA of word to be done
103C DF F0      NEXT3  STX    W
103E EE 00             LDX    0,X        get VECT which points to executable code
                *                                                             =
                * The next instruction could be patched to JMP TRACE          =
                * if a TRACE routine is available:                            =
                *                                                             =
1040 6E 00             JMP    0,X
1042 01                NOP
                *      JMP TRACE ( an alternate for the above )
                *                                                             =
                * = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

```
                *
                * ======>>  1  <<
1043 83                 FCB     $83
1044 4C                 FCC     2,LIT       NOTE: this is different from LITERAL
1046 D4                 FCB     $D4
1047 00 00              FDB     0            link of zero to terminate dictionary scan
1049 10 4B      LIT     FDB     *+2
104B DE F2              LDX     IP
104D 08                 INX
104E 08                 INX
104F DF F2              STX     IP
1051 A6 00              LDA A   0,X
1053 E6 01              LDA B   1,X
1055 7E 10 32           JMP     PUSHBA
                *
                * ######>> screen 14 <<
                * ======>>  2  <<
1058 10 5A      CLITER  FDB     *+2         ( this is an invisible word, with no header )
105A DE F2              LDX     IP
105C 08                 INX
105D DF F2              STX     IP
105F 4F                 CLR A
1060 E6 01              LDA B   1,X
1062 7E 10 32           JMP     PUSHBA
                *
                * ======>>  3  <<
1065 87                 FCB     $87
1066 45                 FCC     6,EXECUTE
106C C5                 FCB     $C5
106D 10 43              FDB     LIT-6
106F 10 71      EXEC    FDB     *+2
1071 30                 TSX
1072 EE 00              LDX     0,X         get code field address (CFA)
1074 31                 INS                 pop stack
1075 31                 INS
1076 7E 10 3C           JMP     NEXT3
                *
                * ######>> screen 15 <<
                * ======>>  4  <<
1079 86                 FCB     $86
107A 42                 FCC     5,BRANCH
107F C8                 FCB     $C8
1080 10 65              FDB     EXEC-10
1082 10 97      BRAN    FDB     ZBYES       Go steal code in ZBRANCH
                *
                * ======>>  5  <<
1084 87                 FCB     $87
1085 30                 FCC     6,0BRANCH
108B C8                 FCB     $C8
108C 10 79              FDB     BRAN-9
108E 10 90      ZBRAN   FDB     *+2
1090 32                 PUL A
1091 33                 PUL B
1092 1B                 ABA
1093 26 13              BNE     ZBNO
```

```
1095 25 11            BCS    ZBNO
1097 DE F2     ZBYES  LDX    IP          Note: code is shared with BRANCH, (+LOOP), (LOOP)
1099 E6 03            LDA B  3,X
109B A6 02            LDA A  2,X
109D DB F3            ADD B  IP+1
109F 99 F2            ADC A  IP
10A1 D7 F3            STA B  IP+1
10A3 97 F2            STA A  IP
10A5 7E 10 34         JMP    NEXT
10A8 DE F2     ZBNO   LDX    IP          no branch. This code is shared with (+LOOP) & (LOOP).
10AA 08               INX                jump over branch delta
10AB 08               INX
10AC DF F2            STX    IP
10AE 7E 10 34         JMP    NEXT
               *
               * ######>> screen 16 <<
               * ======>>  6  <<
10B1 86               FCB    $86
10B2 28               FCC    5,(LOOP)
10B7 A9               FCB    $A9
10B8 10 84            FDB    ZBRAN-10
10BA 10 BC     XLOOP  FDB    *+2
10BC 4F               CLR A
10BD C6 01            LDA B  #1          get set to increment counter by 1
10BF 20 0E            BRA    XPLOP2      go steal other guy's code!
               *
               * ======>>  7  <<
10C1 87               FCB    $87
10C2 28               FCC    6,(+LOOP)
10C8 A9               FCB    $A9
10C9 10 B1            FDB    XLOOP-9
10CB 10 CD     XPLOOP FDB    *+2         Note: +LOOP has an un-signed loop counter
10CD 32               PUL A              get increment
10CE 33               PUL B
10CF 4D        XPLOP2 TST A
10D0 2A 16            BPL    XPLOF       forward looping
10D2 8D 09            BSR    XPLOPS
10D4 0D               SEC
10D5 E2 05            SBC B  5,X
10D7 A2 04            SBC A  4,X
10D9 2A BC            BPL    ZBYES
10DB 20 13            BRA    XPLONO      fall thru
               *
               *  the subroutine :
10DD DE F4     XPLOPS LDX    RP
10DF EB 03            ADD B  3,X         add it to counter
10E1 A9 02            ADC A  2,X
10E3 E7 03            STA B  3,X         store new counter value
10E5 A7 02            STA A  2,X
10E7 39               RTS
               *
10E8 8D F3     XPLOF  BSR    XPLOPS
10EA E0 05            SUB B  5,X
10EC A2 04            SBC A  4,X
10EE 2B A7            BMI    ZBYES
               *
```

```
10F0 08        XPLONO INX              done, don't branch back
10F1 08               INX
10F2 08               INX
10F3 08               INX
10F4 DF F4            STX    RP
10F6 20 B0            BRA    ZBNO     use ZBRAN to skip over unused delta
               *
               * ######>> screen 17 <<
               * ======>>  8  <<
10F8 84               FCB    $84
10F9 28               FCC    3,(DO)
10FC A9               FCB    $A9
10FD 10 C1            FDB    XPLOOP-10
10FF 11 01     XDO    FDB    *+2      This is the RUN-TIME DO, not the COMPILING DO
1101 DE F4            LDX    RP
1103 09               DEX
1104 09               DEX
1105 09               DEX
1106 09               DEX
1107 DF F4            STX    RP
1109 32               PUL A
110A 33               PUL B
110B A7 02            STA A  2,X
110D E7 03            STA B  3,X
110F 32               PUL A
1110 33               PUL B
1111 A7 04            STA A  4,X
1113 E7 05            STA B  5,X
1115 7E 10 34         JMP    NEXT
               *
               * ======>>  9  <<
1118 81               FCB    $81      I
1119 C9               FCB    $C9
111A 10 F8            FDB    XDO-7
111C 11 1E     I      FDB    *+2
111E DE F4            LDX    RP
1120 08               INX
1121 08               INX
1122 7E 10 2E         JMP    GETX
               *
               * ######>> screen 18 <<
               * ======>>  10  <<
1125 85               FCB    $85
1126 44               FCC    4,DIGIT
112A D4               FCB    $D4
112B 11 18            FDB    I-4
112D 11 2F     DIGIT  FDB    *+2      NOTE: legal input range is 0-9, A-Z
112F 30               TSX
1130 A6 03            LDA A  3,X
1132 80 30            SUB A  #$30     ascii zero
1134 2B 1B            BMI    DIGIT2   IF LESS THAN '0', ILLEGAL
1136 81 0A            CMP A  #$A
1138 2B 0A            BMI    DIGIT0   IF '9' OR LESS
113A 81 11            CMP A  #$11
113C 2B 13            BMI    DIGIT2   if less than "A"
113E 81 2B            CMP A  #$2B
```

```
1140 2A 0F            BPL    DIGIT2     if greater than "Z"
1142 80 07            SUB A  #7         translate 'A' thru 'F'
1144 A1 01     DIGIT0 CMP A  1,X
1146 2A 09            BPL    DIGIT2     if not less than the base
1148 C6 01            LDA B  #1         set flag
114A A7 03            STA A  3,X        store digit
114C E7 01     DIGIT1 STA B  1,X        store the flag
114E 7E 10 34         JMP    NEXT
1151 5F        DIGIT2 CLR B
1152 31               INS
1153 31               INS                pop bottom number
1154 30               TSX
1155 E7 00            STA B  0,X        make sure both bytes are 00
1157 20 F3            BRA    DIGIT1
                *
                * ######>> screen 19 <<
                *
                * The word format in the dictionary is:
                *
                * char-count + $80     lowest address
                * char 1
                * char 2
                *
                * char n  + $80
                * link high byte \___point to previous word
                * link low  byte /
                * CFA  high byte \___pnt to 6800 code
                * CFA  low  byte /
                * parameter fields
                *    "
                *    "
                *    "
                *
                * ======>>  11  <<
1159 86               FCB    $86
115A 28               FCC    5,(FIND)
115F A9               FCB    $A9
1160 11 25            FDB    DIGIT-8
1162 11 64     PFIND  FDB    *+2
1164 01               NOP
1165 01               NOP
00E0           PD     EQU    N          ptr to dict word being checked
00E2           PA0    EQU    N+2
00E4           PA     EQU    N+4
00E6           PC     EQU    N+6
1166 CE 00 E0         LDX    #PD
1169 C6 04            LDA B  #4
116B 32        PFIND0 PUL A              loop to get arguments
116C A7 00            STA A  0,X
116E 08               INX
116F 5A               DEC B
1170 26 F9            BNE    PFIND0
                *
1172 DE E0            LDX    PD
1174 E6 00     PFIND1 LDA B  0,X         get count dict count
1176 D7 E6            STA B  PC
```

```
1178 C4 3F            AND B  #$3F
117A 08               INX
117B DF E0            STX    PD          update PD
117D DE E2            LDX    PA0
117F A6 00            LDA A  0,X         get count from arg
1181 08               INX
1182 DF E4            STX    PA          initialize PA
1184 11               CBA                compare lengths
1185 26 22            BNE    PFIND4
1187 DE E4     PFIND2 LDX    PA
1189 A6 00            LDA A  0,X
118B 08               INX
118C DF E4            STX    PA
118E DE E0            LDX    PD
1190 E6 00            LDA B  0,X
1192 08               INX
1193 DF E0            STX    PD
1195 5D               TST B              is dict entry neg. ?
1196 2A 0E            BPL    PFIND8
1198 C4 7F            AND B  #$7F        clear sign
119A 11               CBA
119B 27 15            BEQ    FOUND
119D EE 00     PFIND3 LDX    0,X         get new link
119F 26 D3            BNE    PFIND1      continue if link not=0
               *
               *     not found :
               *
11A1 4F               CLR A
11A2 5F               CLR B
11A3 7E 10 32         JMP    PUSHBA
11A6 11        PFIND8 CBA
11A7 27 DE            BEQ    PFIND2
11A9 DE E0     PFIND4 LDX    PD
11AB E6 00     PFIND9 LDA B  0,X         scan forward to end of this name
11AD 08               INX
11AE 2A FB            BPL    PFIND9
11B0 20 EB            BRA    PFIND3
               *
               *      found :
               *
11B2 96 E0     FOUND  LDA A  PD          compute CFA
11B4 D6 E1            LDA B  PD+1
11B6 CB 04            ADD B  #4
11B8 89 00            ADC A  #0
11BA 37               PSH B
11BB 36               PSH A
11BC 96 E6            LDA A  PC
11BE 36               PSH A
11BF 4F               CLR A
11C0 36               PSH A
11C1 C6 01            LDA B  #1
11C3 7E 10 32         JMP    PUSHBA
               *
11C6 36               PSH A
11C7 4F               CLR A
11C8 36               PSH A
```

```
11C9 C6 01          LDA B  #1
11CB 7E 10 32       JMP    PUSHBA
                *
                * ######>> screen 20 <<
                * ======>>  12  <<
11CE 87             FCB    $87
11CF 45             FCC    6,ENCLOSE
11D5 C5             FCB    $C5
11D6 11 59          FDB    PFIND-9
                * NOTE :
                * FC means offset (bytes) to First Character of next word
                * EW  "      "   to End of Word
                * NC  "      "   to Next Character to start next enclose at
11D8 11 DA      ENCLOS FDB    *+2
11DA 31             INS
11DB 33             PUL B             now,get the low byte, for an 8-bit delimiter
11DC 30             TSX
11DD EE 00          LDX    0,X
11DF 7F 00 E0       CLR    N
                *       wait for a non-delimiter or a NUL
11E2 A6 00      ENCL2  LDA A  0,X
11E4 27 24          BEQ    ENCL6
11E6 11             CBA                CHECK FOR DELIM
11E7 26 06          BNE    ENCL3
11E9 08             INX
11EA 7C 00 E0       INC    N
11ED 20 F3          BRA    ENCL2
                *       found first charcter. Push FC
11EF 96 E0      ENCL3  LDA A  N           found first char.
11F1 36             PSH A
11F2 4F             CLR A
11F3 36             PSH A
                *       wait for a delimiter or a NUL
11F4 A6 00      ENCL4  LDA A  0,X
11F6 27 19          BEQ    ENCL7
11F8 11             CBA                ckech for delim.
11F9 27 06          BEQ    ENCL5
11FB 08             INX
11FC 7C 00 E0       INC    N
11FF 20 F3          BRA    ENCL4
                *       found EW. Push it
1201 D6 E0      ENCL5  LDA B  N
1203 4F             CLR A
1204 37             PSH B
1205 36             PSH A
                *       advance and push NC
1206 5C             INC B
1207 7E 10 32       JMP    PUSHBA
                *   found NUL before non-delimiter, therefore there is no word
120A D6 E0      ENCL6  LDA B  N           found NUL
120C 37             PSH B
120D 36             PSH A
120E 5C             INC B
120F 20 02          BRA    ENCL7+2
                *       found NUL following the word instead of SPACE
1211 D6 E0      ENCL7  LDA B  N
```

```
 1213 37              PSH B          save EW
 1214 36              PSH A
 1215 D6 E0     ENCL8 LDA B  N       save NC
 1217 7E 10 32        JMP    PUSHBA
```

```
                *
                * ######>> screen 21 <<
                * The next 4 words call system-dependant I/O subroutines
                * which are listed after word "-->" ( lable: "arrow" )
                * in the dictionary.
                *
                * ======>>  13  <<
121A 84                 FCB    $84
121B 45                 FCC    3,EMIT
121E D4                 FCB    $D4
121F 11 CE              FDB    ENCLOS-10
1221 12 23      EMIT    FDB    *+2
1223 32                 PUL A
1224 32                 PUL A
1225 BD 23 00           JSR    PEMIT
1228 DE F6              LDX    UP
122A 6C 1B              INC    XOUT+1-UORIG,X
122C 26 02              BNE    *+4
122E 6C 1A              INC    XOUT-UORIG,X
1230 7E 10 34           JMP    NEXT
                *
                * ======>>  14  <<
1233 83                 FCB    $83
1234 4B                 FCC    2,KEY
1236 D9                 FCB    $D9
1237 12 1A              FDB    EMIT-7
1239 12 3B      KEY     FDB    *+2
123B BD 23 17           JSR    PKEY
123E 36                 PSH A
123F 4F                 CLR A
1240 36                 PSH A
1241 7E 10 34           JMP    NEXT
                *
                * ======>>  15  <<
1244 89                 FCB    $89
1245 3F                 FCC    8,?TERMINAL
124D CC                 FCB    $CC
124E 12 33              FDB    KEY-6
1250 12 52      QTERM   FDB    *+2
1252 BD 23 2F           JSR    PQTER
1255 5F                 CLR B
1256 7E 10 32           JMP    PUSHBA      stack the flag
                *
                * ======>>  16  <<
1259 82                 FCB    $82
125A 43                 FCC    1,CR
125B D2                 FCB    $D2
125C 12 44              FDB    QTERM-12
125E 12 60      CR      FDB    *+2
1260 BD 23 3C           JSR    PCR
1263 7E 10 34           JMP    NEXT
                *
                * ######>> screen 22 <<
                * ======>>  17  <<
1266 85                 FCB    $85
```

```
1267 43                FCC    4,CMOVE source, destination, count
126B C5                FCB    $C5
126C 12 59             FDB    CR-5
126E 12 70     CMOVE   FDB    *+2         takes ( 43+47*count ) cycles
1270 CE 00 E0          LDX    #N
1273 C6 06             LDA B  #6
1275 32        CMOV1   PUL A
1276 A7 00             STA A  0,X         move parameters to scratch area
1278 08                INX
1279 5A                DEC B
127A 26 F9             BNE    CMOV1
127C 96 E0     CMOV2   LDA A  N
127E D6 E1             LDA B  N+1
1280 C0 01             SUB B  #1
1282 82 00             SBC A  #0
1284 97 E0             STA A  N
1286 D7 E1             STA B  N+1
1288 25 10             BCS    CMOV3
128A DE E4             LDX    N+4
128C A6 00             LDA A  0,X
128E 08                INX
128F DF E4             STX    N+4
1291 DE E2             LDX    N+2
1293 A7 00             STA A  0,X
1295 08                INX
1296 DF E2             STX    N+2
1298 20 E2             BRA    CMOV2
129A 7E 10 34  CMOV3   JMP    NEXT
               *
               * ######>> screen 23 <<
               * ======>>  18  <<
129D 82                FCB    $82
129E 55                FCC    1,U*
129F AA                FCB    $AA
12A0 12 66             FDB    CMOVE-8
12A2 12 A4     USTAR   FDB    *+2
12A4 8D 05             BSR    USTARS
12A6 31                INS
12A7 31                INS
12A8 7E 10 32          JMP    PUSHBA
               *
               *  The following is a subroutine which
               *  multiplies top 2 words on stack,
               *  leaving 32-bit result:  high order word in A,B
               *  low order word in 2nd word of stack.
               *
12AB 86 10     USTARS  LDA A  #16         bits/word counter
12AD 36                PSH A
12AE 4F                CLR A
12AF 5F                CLR B
12B0 30                TSX
12B1 66 05     USTAR2  ROR    5,X         shift multiplier
12B3 66 06             ROR    6,X
12B5 6A 00             DEC    0,X         done?
12B7 2B 0A             BMI    USTAR4
12B9 24 04             BCC    USTAR3
```

```
12BB EB 04            ADD B  4,X
12BD A9 03            ADC A  3,X
12BF 46        USTAR3 ROR A
12C0 56               ROR B              shift result
12C1 20 EE            BRA    USTAR2
12C3 31        USTAR4 INS                dump counter
12C4 39               RTS
               *
               * ######>> screen 24 <<
               * ======>>  19  <<
12C5 82               FCB    $82
12C6 55               FCC    1,U/
12C7 AF               FCB    $AF
12C8 12 9D            FDB    USTAR-5
12CA 12 CC     USLASH FDB    *+2
12CC 86 11            LDA A  #17
12CE 36               PSH A
12CF 30               TSX
12D0 A6 03            LDA A  3,X
12D2 E6 04            LDA B  4,X
12D4 A1 01     USL1   CMP A  1,X
12D6 22 09            BHI    USL3
12D8 25 04            BCS    USL2
12DA E1 02            CMP B  2,X
12DC 24 03            BCC    USL3
12DE 0C        USL2   CLC
12DF 20 05            BRA    USL4
12E1 E0 02     USL3   SUB B  2,X
12E3 A2 01            SBC A  1,X
12E5 0D               SEC
12E6 69 06     USL4   ROL    6,X
12E8 69 05            ROL    5,X
12EA 6A 00            DEC    0,X
12EC 27 06            BEQ    USL5
12EE 59               ROL B
12EF 49               ROL A
12F0 24 E2            BCC    USL1
12F2 20 ED            BRA    USL3
12F4 31        USL5   INS
12F5 31               INS
12F6 31               INS
12F7 31               INS
12F8 31               INS
12F9 7E 14 7E         JMP    SWAP+4     reverse quotient & remainder
               *
               * ######>> screen 25 <<
               * ======>>  20  <<
12FC 83               FCB    $83
12FD 41               FCC    2,AND
12FF C4               FCB    $C4
1300 12 C5            FDB    USLASH-5
1302 13 04     AND    FDB    *+2
1304 32               PUL A
1305 33               PUL B
1306 30               TSX
1307 E4 01            AND B  1,X
```

```
1309 A4 00            AND A  0,X
130B 7E 10 28         JMP    STABX
                *
                * ======>>  21  <<
130E 82               FCB    $82
130F 4F               FCC    1,OR
1310 D2               FCB    $D2
1311 12 FC            FDB    AND-6
1313 13 15      OR    FDB    *+2
1315 32               PUL A
1316 33               PUL B
1317 30               TSX
1318 EA 01            ORA B  1,X
131A AA 00            ORA A  0,X
131C 7E 10 28         JMP    STABX
                *
                * ======>>  22  <<
131F 83               FCB    $83
1320 58               FCC    2,XOR
1322 D2               FCB    $D2
1323 13 0E            FDB    OR-5
1325 13 27      XOR   FDB    *+2
1327 32               PUL A
1328 33               PUL B
1329 30               TSX
132A E8 01            EOR B  1,X
132C A8 00            EOR A  0,X
132E 7E 10 28         JMP    STABX
                *
                * ######>> screen 26 <<
                * ======>>  23  <<
1331 83               FCB    $83
1332 53               FCC    2,SP@
1334 C0               FCB    $C0
1335 13 1F            FDB    XOR-6
1337 13 39      SPAT  FDB    *+2
1339 30               TSX
133A DF E0            STX    N          scratch area
133C CE 00 E0         LDX    #N
133F 7E 10 2E         JMP    GETX
                *
                * ======>>  24  <<
1342 83               FCB    $83
1343 53               FCC    2,SP!
1345 A1               FCB    $A1
1346 13 31            FDB    SPAT-6
1348 13 4A      SPSTOR FDB   *+2
134A DE F6            LDX    UP
134C EE 06            LDX    XSPZER-UORIG,X
134E 35               TXS              watch it ! X and S are not equal.
134F 7E 10 34         JMP    NEXT
                *
                * ======>>  25  <<
1352 83               FCB    $83
1353 52               FCC    2,RP!
1355 A1               FCB    $A1
```

```
1356 13 42            FDB    SPSTOR-6
1358 13 5A     RPSTOR FDB    *+2
135A FE 10 14         LDX    RINIT      initialize from rom constant
135D DF F4            STX    RP
135F 7E 10 34         JMP    NEXT
               *
               * ======>>  26  <<
1362 82               FCB    $82
1363 3B               FCC    1,;S
1364 D3               FCB    $D3
1365 13 52            FDB    RPSTOR-6
1367 13 69     SEMIS  FDB    *+2
1369 DE F4            LDX    RP
136B 08               INX
136C 08               INX
136D DF F4            STX    RP
136F EE 00            LDX    0,X        get address we have just finished.
1371 7E 10 36         JMP    NEXT+2     increment the return address & do next word
               *
               * ######>> screen 27 <<
               * ======>>  27  <<
1374 85               FCB    $85
1375 4C               FCC    4,LEAVE
1379 C5               FCB    $C5
137A 13 62            FDB    SEMIS-5
137C 13 7E     LEAVE  FDB    *+2
137E DE F4            LDX    RP
1380 A6 02            LDA A  2,X
1382 E6 03            LDA B  3,X
1384 A7 04            STA A  4,X
1386 E7 05            STA B  5,X
1388 7E 10 34         JMP    NEXT
               *
               * ======>>  28  <<
138B 82               FCB    $82
138C 3E               FCC    1,>R
138D D2               FCB    $D2
138E 13 74            FDB    LEAVE-8
1390 13 92     TOR    FDB    *+2
1392 DE F4            LDX    RP
1394 09               DEX
1395 09               DEX
1396 DF F4            STX    RP
1398 32               PUL A
1399 33               PUL B
139A A7 02            STA A  2,X
139C E7 03            STA B  3,X
139E 7E 10 34         JMP    NEXT
               *
               * ======>>  29  <<
13A1 82               FCB    $82
13A2 52               FCC    1,R>
13A3 BE               FCB    $BE
13A4 13 8B            FDB    TOR-5
13A6 13 A8     FROMR  FDB    *+2
13A8 DE F4            LDX    RP
```

```
13AA A6 02          LDA A  2,X
13AC E6 03          LDA B  3,X
13AE 08             INX
13AF 08             INX
13B0 DF F4          STX    RP
13B2 7E 10 32       JMP    PUSHBA
                    *
                    * ======>>  30  <<
13B5 81             FCB    $81          R
13B6 D2             FCB    $D2
13B7 13 A1          FDB    FROMR-5
13B9 13 BB    R     FDB    *+2
13BB DE F4          LDX    RP
13BD 08             INX
13BE 08             INX
13BF 7E 10 2E       JMP    GETX
                    *
                    * ######>> screen 28 <<
                    * ======>>  31  <<
13C2 82             FCB    $82
13C3 30             FCC    1,0=
13C4 BD             FCB    $BD
13C5 13 B5          FDB    R-4
13C7 13 C9    ZEQU  FDB    *+2
13C9 30             TSX
13CA 4F             CLR A
13CB 5F             CLR B
13CC EE 00          LDX    0,X
13CE 26 01          BNE    ZEQU2
13D0 5C             INC B
13D1 30       ZEQU2 TSX
13D2 7E 10 28       JMP    STABX
                    *
                    * ======>>  32  <<
13D5 82             FCB    $82
13D6 30             FCC    1,0<
13D7 BC             FCB    $BC
13D8 13 C2          FDB    ZEQU-5
13DA 13 DC    ZLESS FDB    *+2
13DC 30             TSX
13DD 86 80          LDA A  #$80         check the sign bit
13DF A4 00          AND A  0,X
13E1 27 06          BEQ    ZLESS2
13E3 4F             CLR A               if neg.
13E4 C6 01          LDA B  #1
13E6 7E 10 28       JMP    STABX
13E9 5F       ZLESS2 CLR B
13EA 7E 10 28       JMP    STABX
                    *
                    * ######>> screen 29 <<
                    * ======>>  33  <<
13ED 81             FCB    $81          +
13EE AB             FCB    $AB
13EF 13 D5          FDB    ZLESS-5
13F1 13 F3    PLUS  FDB    *+2
13F3 32             PUL A
```

```
13F4 33              PUL B
13F5 30              TSX
13F6 EB 01           ADD B  1,X
13F8 A9 00           ADC A  0,X
13FA 7E 10 28        JMP    STABX
                *
                * ======>>  34  <<
13FD 82              FCB    $82
13FE 44              FCC    1,D+
13FF AB              FCB    $AB
1400 13 ED           FDB    PLUS-4
1402 14 04     DPLUS FDB    *+2
1404 30              TSX
1405 0C              CLC
1406 C6 04           LDA B  #4
1408 A6 03     DPLUS2 LDA A  3,X
140A A9 07           ADC A  7,X
140C A7 07           STA A  7,X
140E 09              DEX
140F 5A              DEC B
1410 26 F6           BNE    DPLUS2
1412 31              INS
1413 31              INS
1414 31              INS
1415 31              INS
1416 7E 10 34        JMP    NEXT
                *
                * ======>>  35  <<
1419 85              FCB    $85
141A 4D              FCC    4,MINUS
141E D3              FCB    $D3
141F 13 FD           FDB    DPLUS-5
1421 14 23     MINUS FDB    *+2
1423 30              TSX
1424 60 01           NEG    1,X
1426 25 04           BCS    MINUS2
1428 60 00           NEG    0,X
142A 20 02           BRA    MINUS3
142C 63 00     MINUS2 COM    0,X
142E 7E 10 34  MINUS3 JMP    NEXT
                *
                * ======>>  36  <<
1431 86              FCB    $86
1432 44              FCC    5,DMINUS
1437 D3              FCB    $D3
1438 14 19           FDB    MINUS-8
143A 14 3C     DMINUS FDB    *+2
143C 30              TSX
143D 63 00           COM    0,X
143F 63 01           COM    1,X
1441 63 02           COM    2,X
1443 60 03           NEG    3,X
1445 26 0A           BNE    DMINX
1447 6C 02           INC    2,X
1449 26 06           BNE    DMINX
144B 6C 01           INC    1,X
```

```
144D 26 02            BNE    DMINX
144F 6C 00            INC    0,X
1451 7E 10 34  DMINX  JMP    NEXT
               *
               * ######>> screen 30 <<
               * ======>>  37  <<
1454 84               FCB    $84
1455 4F               FCC    3,OVER
1458 D2               FCB    $D2
1459 14 31            FDB    DMINUS-9
145B 14 5D     OVER   FDB    *+2
145D 30               TSX
145E A6 02            LDA A  2,X
1460 E6 03            LDA B  3,X
1462 7E 10 32         JMP    PUSHBA
               *
               * ======>>  38  <<
1465 84               FCB    $84
1466 44               FCC    3,DROP
1469 D0               FCB    $D0
146A 14 54            FDB    OVER-7
146C 14 6E     DROP   FDB    *+2
146E 31               INS
146F 31               INS
1470 7E 10 34         JMP    NEXT
               *
               * ======>>  39  <<
1473 84               FCB    $84
1474 53               FCC    3,SWAP
1477 D0               FCB    $D0
1478 14 65            FDB    DROP-7
147A 14 7C     SWAP   FDB    *+2
147C 32               PUL A
147D 33               PUL B
147E 30               TSX
147F EE 00            LDX    0,X
1481 31               INS
1482 31               INS
1483 37               PSH B
1484 36               PSH A
1485 DF E0            STX    N
1487 CE 00 E0         LDX    #N
148A 7E 10 2E         JMP    GETX
               *
               * ======>>  40  <<
148D 83               FCB    $83
148E 44               FCC    2,DUP
1490 D0               FCB    $D0
1491 14 73            FDB    SWAP-7
1493 14 95     DUP    FDB    *+2
1495 32               PUL A
1496 33               PUL B
1497 37               PSH B
1498 36               PSH A
1499 7E 10 32         JMP    PUSHBA
               *
```

```
                * ######>> screen 31 <<
                * ======>>  41  <<
149C 82               FCB    $82
149D 2B               FCC    1,+!
149E A1               FCB    $A1
149F 14 8D            FDB    DUP-6
14A1 14 A3      PSTORE FDB   *+2
14A3 30               TSX
14A4 EE 00            LDX    0,X
14A6 31               INS
14A7 31               INS
14A8 32               PUL A             get stack data
14A9 33               PUL B
14AA EB 01            ADD B  1,X        add & store low byte
14AC E7 01            STA B  1,X
14AE A9 00            ADC A  0,X        add & store hi byte
14B0 A7 00            STA A  0,X
14B2 7E 10 34         JMP    NEXT
                *
                * ======>>  42  <<
14B5 86               FCB    $86
14B6 54               FCC    5,TOGGLE
14BB C5               FCB    $C5
14BC 14 9C            FDB    PSTORE-5
14BE 15 25      TOGGLE FDB   DOCOL,OVER,CAT,XOR,SWAP,CSTORE
14CA 13 67            FDB    SEMIS
                *
                * ######>> screen 32 <<
                * ======>>  43  <<
14CC 81               FCB    $81        @
14CD C0               FCB    $C0
14CE 14 B5            FDB    TOGGLE-9
14D0 14 D2      AT     FDB   *+2
14D2 30               TSX
14D3 EE 00            LDX    0,X        get address
14D5 31               INS
14D6 31               INS
14D7 7E 10 2E         JMP    GETX
                *
                * ======>>  44  <<
14DA 82               FCB    $82
14DB 43               FCC    1,C@
14DC C0               FCB    $C0
14DD 14 CC            FDB    AT-4
14DF 14 E1      CAT    FDB   *+2
14E1 30               TSX
14E2 EE 00            LDX    0,X
14E4 4F               CLR A
14E5 E6 00            LDA B  0,X
14E7 31               INS
14E8 31               INS
14E9 7E 10 32         JMP    PUSHBA
                *
                * ======>>  45  <<
14EC 81               FCB    $81        !
14ED A1               FCB    $A1
```

```
14EE 14 DA            FDB    CAT-5
14F0 14 F2     STORE  FDB    *+2
14F2 30               TSX
14F3 EE 00            LDX    0,X        get address
14F5 31               INS
14F6 31               INS
14F7 7E 10 26         JMP    PULABX
               *
               * ======>>  46  <<
14FA 82               FCB    $82
14FB 43               FCC    1,C!
14FC A1               FCB    $A1
14FD 14 EC            FDB    STORE-4
14FF 15 01     CSTORE FDB    *+2
1501 30               TSX
1502 EE 00            LDX    0,X
1504 31               INS
1505 31               INS
1506 31               INS
1507 33               PUL B
1508 E7 00            STA B  0,X
150A 7E 10 34         JMP    NEXT
```

```
                *
                * ######>> screen 33 <<
                * ======>>  47  <<
150D C1               FCB    $C1         :   immediate
150E BA               FCB    $BA
150F 14 FA            FDB    CSTORE-5
1511 15 25      COLON FDB    DOCOL,QEXEC,SCSP,CURENT,AT,CONTXT,STORE
151F 1D D1            FDB    CREATE,RBRAK
1523 19 A5            FDB    PSCODE



                * Here is the IP pusher for allowing
                * nested words in the virtual machine:
                *
                * ( ;S  is the equivalent un-nester )



1525 DE F4      DOCOL LDX    RP          make room in the stack
1527 09               DEX
1528 09               DEX
1529 DF F4            STX    RP
152B 96 F2            LDA A  IP
152D D6 F3            LDA B  IP+1
152F A7 02            STA A  2,X         Store address of the high level word
1531 E7 03            STA B  3,X         that we are starting to execute
1533 DE F0            LDX    W           Get first sub-word of that definition
1535 7E 10 36         JMP    NEXT+2      and execute it
                *
                * ======>>  48  <<
1538 C1               FCB    $C1         ;   immediate code
1539 BB               FCB    $BB
153A 15 0D            FDB    COLON-4
153C 15 25      SEMI  FDB    DOCOL,QCSP,COMPIL,SEMIS,SMUDGE,LBRAK
1548 13 67            FDB    SEMIS
                *
                * ######>> screen 34 <<
                * ======>>  49  <<
154A 88               FCB    $88
154B 43               FCC    7,CONSTANT
1552 D4               FCB    $D4
1553 15 38            FDB    SEMI-4
1555 15 25      CON   FDB    DOCOL,CREATE,SMUDGE,COMMA,PSCODE
155F DE F0      DOCON LDX    W
1561 A6 02            LDA A  2,X
1563 E6 03            LDA B  3,X         A & B now contain the constant
1565 7E 10 32         JMP    PUSHBA
                *
                * ======>>  50  <<
1568 88               FCB    $88
```

```
1569 56                 FCC    7,VARIABLE
1570 C5                 FCB    $C5
1571 15 4A              FDB    CON-11
1573 15 25      VAR     FDB    DOCOL,CON,PSCODE
1579 96 F0      DOVAR   LDA A  W
157B D6 F1              LDA B  W+1
157D CB 02              ADD B  #2
157F 89 00              ADC A  #0          A,B now contain the address of the variable
1581 7E 10 32           JMP    PUSHBA
                *
                * ======>>  51  <<
1584 84                 FCB    $84
1585 55                 FCC    3,USER
1588 D2                 FCB    $D2
1589 15 68              FDB    VAR-11
158B 15 25      USER    FDB    DOCOL,CON,PSCODE
1591 DE F0      DOUSER  LDX    W           get offset into user's table
1593 A6 02              LDA A  2,X
1595 E6 03              LDA B  3,X
1597 DB F7              ADD B  UP+1        add to users base address
1599 99 F6              ADC A  UP
159B 7E 10 32           JMP    PUSHBA      push address of user's variable
                *
                * ######>> screen 35 <<
                * ======>>  52  <<
159E 81                 FCB    $81         0
159F B0                 FCB    $B0
15A0 15 84              FDB    USER-7
15A2 15 5F      ZERO    FDB    DOCON
15A4 00 00              FDB    0000
                *
                * ======>>  53  <<
15A6 81                 FCB    $81         1
15A7 B1                 FCB    $B1
15A8 15 9E              FDB    ZERO-4
15AA 15 5F      ONE     FDB    DOCON
15AC 00 01              FDB    1
                *
                * ======>>  54  <<
15AE 81                 FCB    $81         2
15AF B2                 FCB    $B2
15B0 15 A6              FDB    ONE-4
15B2 15 5F      TWO     FDB    DOCON
15B4 00 02              FDB    2
                *
                * ======>>  55  <<
15B6 81                 FCB    $81         3
15B7 B3                 FCB    $B3
15B8 15 AE              FDB    TWO-4
15BA 15 5F      THREE   FDB    DOCON
15BC 00 03              FDB    3
                *
                * ======>>  56  <<
15BE 82                 FCB    $82
15BF 42                 FCC    1,BL
15C0 CC                 FCB    $CC
```

```
15C1 15 B6              FDB    THREE-4
15C3 15 5F     BL       FDB    DOCON       ascii blank
15C5 00 20              FDB    $20
               *
               * ======>>  57  <<
15C7 85                 FCB    $85
15C8 46                 FCC    4,FIRST
15CC D4                 FCB    $D4
15CD 15 BE              FDB    BL-5
15CF 15 5F     FIRST    FDB    DOCON
15D1 30 00              FDB    MEMEND-528 (132*NBLK)
               *
               * ======>>  58  <<
15D3 85                 FCB    $85
15D4 4C                 FCC    4,LIMIT ( the end of memory +1 )
15D8 D4                 FCB    $D4
15D9 15 C7              FDB    FIRST-8
15DB 15 5F     LIMIT    FDB    DOCON
15DD 32 10              FDB    MEMEND
               *
               * ======>>  59  <<
15DF 85                 FCB    $85
15E0 42                 FCC    4,B/BUF (bytes/buffer)
15E4 C6                 FCB    $C6
15E5 15 D3              FDB    LIMIT-8
15E7 15 5F     BBUF     FDB    DOCON
15E9 00 80              FDB    128
               *
               * ======>>  60  <<
15EB 85                 FCB    $85
15EC 42                 FCC    4,B/SCR (blocks/screen)
15F0 D2                 FCB    $D2
15F1 15 DF              FDB    BBUF-8
15F3 15 5F     BSCR     FDB    DOCON
15F5 00 08              FDB    8
               *  blocks/screen = 1024 / "B/BUF" = 8
               *
               * ======>>  61  <<
15F7 87                 FCB    $87
15F8 2B                 FCC    6,+ORIGIN
15FE CE                 FCB    $CE
15FF 15 EB              FDB    BSCR-8
1601 15 25     PORIG    FDB    DOCOL,LIT,ORIG,PLUS
1609 13 67              FDB    SEMIS
               *
               * ######>> screen 36 <<
               * ======>>  62  <<
160B 82                 FCB    $82
160C 53                 FCC    1,S0
160D B0                 FCB    $B0
160E 15 F7              FDB    PORIG-10
1610 15 91     SZERO    FDB    DOUSER
1612 00 06              FDB    XSPZER-UORIG
               *
               * ======>>  63  <<
1614 82                 FCB    $82
```

```
1615 52              FCC   1,R0
1616 B0              FCB   $B0
1617 16 0B           FDB   SZERO-5
1619 15 91     RZERO FDB   DOUSER
161B 00 08           FDB   XRZERO-UORIG
                     *
                     * ======>> 64 <<
161D 83              FCB   $83
161E 54              FCC   2,TIB
1620 C2              FCB   $C2
1621 16 14           FDB   RZERO-5
1623 15 91     TIB   FDB   DOUSER
1625 00 0A           FDB   XTIB-UORIG
                     *
                     * ======>> 65 <<
1627 85              FCB   $85
1628 57              FCC   4,WIDTH
162C C8              FCB   $C8
162D 16 1D           FDB   TIB-6
162F 15 91     WIDTH FDB   DOUSER
1631 00 0C           FDB   XWIDTH-UORIG
                     *
                     * ======>> 66 <<
1633 87              FCB   $87
1634 57              FCC   6,WARNING
163A C7              FCB   $C7
163B 16 27           FDB   WIDTH-8
163D 15 91     WARN  FDB   DOUSER
163F 00 0E           FDB   XWARN-UORIG
                     *
                     * ======>> 67 <<
1641 85              FCB   $85
1642 46              FCC   4,FENCE
1646 C5              FCB   $C5
1647 16 33           FDB   WARN-10
1649 15 91     FENCE FDB   DOUSER
164B 00 10           FDB   XFENCE-UORIG
                     *
                     * ======>> 68 <<
164D 82              FCB   $82
164E 44              FCC   1,DP  points to first free byte at end of dictionary
164F D0              FCB   $D0
1650 16 41           FDB   FENCE-8
1652 15 91     DP    FDB   DOUSER
1654 00 12           FDB   XDP-UORIG
                     *
                     * ======>> 68.5 <<
1656 88              FCB   $88
1657 56              FCC   7,VOC-LINK
165E CB              FCB   $CB
165F 16 4D           FDB   DP-5
1661 15 91     VOCLIN FDB  DOUSER
1663 00 14           FDB   XVOCL-UORIG
                     *
                     * ======>> 69 <<
1665 83              FCB   $83
```

```
1666 42              FCC   2,BLK
1668 CB              FCB   $CB
1669 16 56           FDB   VOCLIN-11
166B 15 91     BLK   FDB   DOUSER
166D 00 16           FDB   XBLK-UORIG
                     *
                     * ======>>  70  <<
166F 82              FCB   $82
1670 49              FCC   1,IN scan pointer for input line buffer
1671 CE              FCB   $CE
1672 16 65           FDB   BLK-6
1674 15 91     IN    FDB   DOUSER
1676 00 18           FDB   XIN-UORIG
                     *
                     * ======>>  71  <<
1678 83              FCB   $83
1679 4F              FCC   2,OUT
167B D4              FCB   $D4
167C 16 6F           FDB   IN-5
167E 15 91     OUT   FDB   DOUSER
1680 00 1A           FDB   XOUT-UORIG
                     *
                     * ======>>  72  <<
1682 83              FCB   $83
1683 53              FCC   2,SCR
1685 D2              FCB   $D2
1686 16 78           FDB   OUT-6
1688 15 91     SCR   FDB   DOUSER
168A 00 1C           FDB   XSCR-UORIG
                     *######>> screen 37 <<
                     *
                     * ======>>  73  <<
168C 86              FCB   $86
168D 4F              FCC   5,OFFSET
1692 D4              FCB   $D4
1693 16 82           FDB   SCR-6
1695 15 91     OFSET FDB   DOUSER
1697 00 1E           FDB   XOFSET-UORIG
                     *
                     * ======>>  74  <<
1699 87              FCB   $87
169A 43              FCC   6,CONTEXT points to pointer to vocab to search first
16A0 D4              FCB   $D4
16A1 16 8C           FDB   OFSET-9
16A3 15 91     CONTXT FDB  DOUSER
16A5 00 20           FDB   XCONT-UORIG
                     *
                     * ======>>  75  <<
16A7 87              FCB   $87
16A8 43              FCC   6,CURRENT points to ptr. to vocab being extended
16AE D4              FCB   $D4
16AF 16 99           FDB   CONTXT-10
16B1 15 91     CURENT FDB  DOUSER
16B3 00 22           FDB   XCURR-UORIG
                     *
                     * ======>>  76  <<
```

```
16B5 85                 FCB    $85
16B6 53                 FCC    4,STATE 1 if compiling ,  0 if not
16BA C5                 FCB    $C5
16BB 16 A7              FDB    CURENT-10
16BD 15 91      STATE   FDB    DOUSER
16BF 00 24              FDB    XSTATE-UORIG
                *
                * ======>>  77  <<
16C1 84                 FCB    $84
16C2 42                 FCC    3,BASE number base for all input & output
16C5 C5                 FCB    $C5
16C6 16 B5              FDB    STATE-8
16C8 15 91      BASE    FDB    DOUSER
16CA 00 26              FDB    XBASE-UORIG
                *
                * ======>>  78  <<
16CC 83                 FCB    $83
16CD 44                 FCC    2,DPL
16CF CC                 FCB    $CC
16D0 16 C1              FDB    BASE-7
16D2 15 91      DPL     FDB    DOUSER
16D4 00 28              FDB    XDPL-UORIG
                *
                * ======>>  79  <<
16D6 83                 FCB    $83
16D7 46                 FCC    2,FLD
16D9 C4                 FCB    $C4
16DA 16 CC              FDB    DPL-6
16DC 15 91      FLD     FDB    DOUSER
16DE 00 2A              FDB    XFLD-UORIG
                *
                * ======>>  80  <<
16E0 83                 FCB    $83
16E1 43                 FCC    2,CSP
16E3 D0                 FCB    $D0
16E4 16 D6              FDB    FLD-6
16E6 15 91      CSP     FDB    DOUSER
16E8 00 2C              FDB    XCSP-UORIG
                *
                * ======>>  81  <<
16EA 82                 FCB    $82
16EB 52                 FCC    1,R#
16EC A3                 FCB    $A3
16ED 16 E0              FDB    CSP-6
16EF 15 91      RNUM    FDB    DOUSER
16F1 00 2E              FDB    XRNUM-UORIG
                *
                * ======>>  82  <<
16F3 83                 FCB    $83
16F4 48                 FCC    2,HLD
16F6 C4                 FCB    $C4
16F7 16 EA              FDB    RNUM-5
16F9 15 5F      HLD     FDB    DOCON
16FB 01 30              FDB    XHLD
                *
                * ======>>  82.5  <<== SPECIAL
```

```
16FD 87                 FCB    $87
16FE 43                 FCC    6,COLUMNS line width of terminal
1704 D3                 FCB    $D3
1705 16 F3              FDB    HLD-6
1707 15 91      COLUMS  FDB    DOUSER
1709 00 34              FDB    XCOLUMS-UORIG
                *
                * ######>> screen 38 <<
                * ======>>  83  <<
170B 82                 FCB    $82
170C 31                 FCC    1,1+
170D AB                 FCB    $AB
170E 16 FD              FDB    COLUMS-10
1710 15 25      ONEP    FDB    DOCOL,ONE,PLUS
1716 13 67              FDB    SEMIS
                *
                * ======>>  84  <<
1718 82                 FCB    $82
1719 32                 FCC    1,2+
171A AB                 FCB    $AB
171B 17 0B              FDB    ONEP-5
171D 15 25      TWOP    FDB    DOCOL,TWO,PLUS
1723 13 67              FDB    SEMIS
                *
                * ======>>  85  <<
1725 84                 FCB    $84
1726 48                 FCC    3,HERE
1729 C5                 FCB    $C5
172A 17 18              FDB    TWOP-5
172C 15 25      HERE    FDB    DOCOL,DP,AT
1732 13 67              FDB    SEMIS
                *
                * ======>>  86  <<
1734 85                 FCB    $85
1735 41                 FCC    4,ALLOT
1739 D4                 FCB    $D4
173A 17 25              FDB    HERE-7
173C 15 25      ALLOT   FDB    DOCOL,DP,PSTORE
1742 13 67              FDB    SEMIS
                *
                * ======>>  87  <<
1744 81                 FCB    $81         , (COMMA)
1745 AC                 FCB    $AC
1746 17 34              FDB    ALLOT-8
1748 15 25      COMMA   FDB    DOCOL,HERE,STORE,TWO,ALLOT
1752 13 67              FDB    SEMIS
                *
                * ======>>  88  <<
1754 82                 FCB    $82
1755 43                 FCC    1,C,
1756 AC                 FCB    $AC
1757 17 44              FDB    COMMA-4
1759 15 25      CCOMM   FDB    DOCOL,HERE,CSTORE,ONE,ALLOT
1763 13 67              FDB    SEMIS
                *
                * ======>>  89  <<
```

```
1765 81              FCB   $81        -
1766 AD              FCB   $AD
1767 17 54           FDB   CCOMM-5
1769 15 25     SUB   FDB   DOCOL,MINUS,PLUS
176F 13 67           FDB   SEMIS
                     *
                     * ======>>  90  <<
1771 81              FCB   $81        =
1772 BD              FCB   $BD
1773 17 65           FDB   SUB-4
1775 15 25     EQUAL FDB   DOCOL,SUB,ZEQU
177B 13 67           FDB   SEMIS
                     *
                     * ======>>  91  <<
177D 81              FCB   $81        <
177E BC              FCB   $BC
177F 17 71           FDB   EQUAL-4
1781 17 83     LESS  FDB   *+2
1783 32              PUL A
1784 33              PUL B
1785 30              TSX
1786 A1 00           CMP A 0,X
1788 31              INS
1789 2E 09           BGT   LESST
178B 26 04           BNE   LESSF
178D E1 01           CMP B 1,X
178F 22 03           BHI   LESST
1791 5F        LESSF CLR B
1792 20 02           BRA   LESSX
1794 C6 01     LESST LDA B #1
1796 4F        LESSX CLR A
1797 31              INS
1798 7E 10 32        JMP   PUSHBA
                     *
                     * ======>>  92  <<
179B 81              FCB   $81        >
179C BE              FCB   $BE
179D 17 7D           FDB   LESS-4
179F 15 25     GREAT FDB   DOCOL,SWAP,LESS
17A5 13 67           FDB   SEMIS
                     *
                     * ======>>  93  <<
17A7 83              FCB   $83
17A8 52              FCC   2,ROT
17AA D4              FCB   $D4
17AB 17 9B           FDB   GREAT-4
17AD 15 25     ROT   FDB   DOCOL,TOR,SWAP,FROMR,SWAP
17B7 13 67           FDB   SEMIS
                     *
                     * ======>>  94  <<
17B9 85              FCB   $85
17BA 53              FCC   4,SPACE
17BE C5              FCB   $C5
17BF 17 A7           FDB   ROT-6
17C1 15 25     SPACE FDB   DOCOL,BL,EMIT
17C7 13 67           FDB   SEMIS
```

```
                *
                * ======>>  95  <<
17C9 83                 FCB    $83
17CA 4D                 FCC    2,MIN
17CC CE                 FCB    $CE
17CD 17 B9              FDB    SPACE-8
17CF 15 25      MIN     FDB    DOCOL,OVER,OVER,GREAT,ZBRAN
17D9 00 04              FDB    MIN2-*
17DB 14 7A              FDB    SWAP
17DD 14 6C      MIN2    FDB    DROP
17DF 13 67              FDB    SEMIS
                *
                * ======>>  96  <<
17E1 83                 FCB    $83
17E2 4D                 FCC    2,MAX
17E4 D8                 FCB    $D8
17E5 17 C9              FDB    MIN-6
17E7 15 25      MAX     FDB    DOCOL,OVER,OVER,LESS,ZBRAN
17F1 00 04              FDB    MAX2-*
17F3 14 7A              FDB    SWAP
17F5 14 6C      MAX2    FDB    DROP
17F7 13 67              FDB    SEMIS
                *
                * ======>>  97  <<
17F9 84                 FCB    $84
17FA 2D                 FCC    3,-DUP
17FD D0                 FCB    $D0
17FE 17 E1              FDB    MAX-6
1800 15 25      DDUP    FDB    DOCOL,DUP,ZBRAN
1806 00 04              FDB    DDUP2-*
1808 14 93              FDB    DUP
180A 13 67      DDUP2   FDB    SEMIS
                *
                * ######>> screen 39 <<
                * ======>>  98  <<
180C 88                 FCB    $88
180D 54                 FCC    7,TRAVERSE
1814 C5                 FCB    $C5
1815 17 F9              FDB    DDUP-7
1817 15 25      TRAV    FDB    DOCOL,SWAP
181B 14 5B      TRAV2   FDB    OVER,PLUS,CLITER
1821 7F                 FCB    $7F
1822 14 5B              FDB    OVER,CAT,LESS,ZBRAN
182A FF F1              FDB    TRAV2-*
182C 14 7A              FDB    SWAP,DROP
1830 13 67              FDB    SEMIS
                *
                * ======>>  99  <<
1832 86                 FCB    $86
1833 4C                 FCC    5,LATEST
1838 D4                 FCB    $D4
1839 18 0C              FDB    TRAV-11
183B 15 25      LATEST  FDB    DOCOL,CURENT,AT,AT
1843 13 67              FDB    SEMIS
                *
                * ======>>  100  <<
```

```
1845 83              FCB   $83
1846 4C              FCC   2,LFA
1848 C1              FCB   $C1
1849 18 32           FDB   LATEST-9
184B 15 25    LFA    FDB   DOCOL,CLITER
184F 04              FCB   4
1850 17 69           FDB   SUB
1852 13 67           FDB   SEMIS
                     *
                     * ======>>  101  <<
1854 83              FCB   $83
1855 43              FCC   2,CFA
1857 C1              FCB   $C1
1858 18 45           FDB   LFA-6
185A 15 25    CFA    FDB   DOCOL,TWO,SUB
1860 13 67           FDB   SEMIS
                     *
                     * ======>>  102  <<
1862 83              FCB   $83
1863 4E              FCC   2,NFA
1865 C1              FCB   $C1
1866 18 54           FDB   CFA-6
1868 15 25    NFA    FDB   DOCOL,CLITER
186C 05              FCB   5
186D 17 69           FDB   SUB,ONE,MINUS,TRAV
1875 13 67           FDB   SEMIS
                     *
                     * ======>>  103  <<
1877 83              FCB   $83
1878 50              FCC   2,PFA
187A C1              FCB   $C1
187B 18 62           FDB   NFA-6
187D 15 25    PFA    FDB   DOCOL,ONE,TRAV,CLITER
1885 05              FCB   5
1886 13 F1           FDB   PLUS
1888 13 67           FDB   SEMIS
                     *
                     * ######>> screen 40 <<
                     * ======>>  104  <<
188A 84              FCB   $84
188B 21              FCC   3,!CSP
188E D0              FCB   $D0
188F 18 77           FDB   PFA-6
1891 15 25    SCSP   FDB   DOCOL,SPAT,CSP,STORE
1899 13 67           FDB   SEMIS
                     *
                     * ======>>  105  <<
189B 86              FCB   $86
189C 3F              FCC   5,?ERROR
18A1 D2              FCB   $D2
18A2 18 8A           FDB   SCSP-7
18A4 15 25    QERR   FDB   DOCOL,SWAP,ZBRAN
18AA 00 08           FDB   QERR2-*
18AC 1D 6C           FDB   ERROR,BRAN
18B0 00 04           FDB   QERR3-*
18B2 14 6C    QERR2  FDB   DROP
```

```
18B4 13 67     QERR3  FDB    SEMIS
               *
               * ======>>  106  <<
18B6 85               FCB    $85
18B7 3F               FCC    4,?COMP
18BB D0               FCB    $D0
18BC 18 9B            FDB    QERR-9
18BE 15 25     QCOMP  FDB    DOCOL,STATE,AT,ZEQU,CLITER
18C8 11               FCB    $11
18C9 18 A4            FDB    QERR
18CB 13 67            FDB    SEMIS
               *
               * ======>>  107  <<
18CD 85               FCB    $85
18CE 3F               FCC    4,?EXEC
18D2 C3               FCB    $C3
18D3 18 B6            FDB    QCOMP-8
18D5 15 25     QEXEC  FDB    DOCOL,STATE,AT,CLITER
18DD 12               FCB    $12
18DE 18 A4            FDB    QERR
18E0 13 67            FDB    SEMIS
               *
               * ======>>  108  <<
18E2 86               FCB    $86
18E3 3F               FCC    5,?PAIRS
18E8 D3               FCB    $D3
18E9 18 CD            FDB    QEXEC-8
18EB 15 25     QPAIRS FDB    DOCOL,SUB,CLITER
18F1 13               FCB    $13
18F2 18 A4            FDB    QERR
18F4 13 67            FDB    SEMIS
               *
               * ======>>  109  <<
18F6 84               FCB    $84
18F7 3F               FCC    3,?CSP
18FA D0               FCB    $D0
18FB 18 E2            FDB    QPAIRS-9
18FD 15 25     QCSP   FDB    DOCOL,SPAT,CSP,AT,SUB,CLITER
1909 14               FCB    $14
190A 18 A4            FDB    QERR
190C 13 67            FDB    SEMIS
               *
               * ======>>  110  <<
190E 88               FCB    $88
190F 3F               FCC    7,?LOADING
1916 C7               FCB    $C7
1917 18 F6            FDB    QCSP-7
1919 15 25     QLOAD  FDB    DOCOL,BLK,AT,ZEQU,CLITER
1923 16               FCB    $16
1924 18 A4            FDB    QERR
1926 13 67            FDB    SEMIS
               *
               * ######>> screen 41 <<
               * ======>>  111  <<
1928 87               FCB    $87
1929 43               FCC    6,COMPILE
```

```
192F C5              FCB   $C5
1930 19 0E           FDB   QLOAD-11
1932 15 25     COMPIL FDB  DOCOL,QCOMP,FROMR,TWOP,DUP,TOR,AT,COMMA
1942 13 67           FDB   SEMIS
                     *
                     * ======>>  112  <<
1944 C1              FCB   $C1        [  immediate
1945 DB              FCB   $DB
1946 19 28           FDB   COMPIL-10
1948 15 25     LBRAK FDB   DOCOL,ZERO,STATE,STORE
1950 13 67           FDB   SEMIS
                     *
                     * ======>>  113  <<
1952 81              FCB   $81        ]
1953 DD              FCB   $DD
1954 19 44           FDB   LBRAK-4
1956 15 25     RBRAK FDB   DOCOL,CLITER
195A C0              FCB   $C0
195B 16 BD           FDB   STATE,STORE
195F 13 67           FDB   SEMIS
                     *
                     * ======>>  114  <<
1961 86              FCB   $86
1962 53              FCC   5,SMUDGE
1967 C5              FCB   $C5
1968 19 52           FDB   RBRAK-4
196A 15 25     SMUDGE FDB  DOCOL,LATEST,CLITER
1970 20              FCB   $20
1971 14 BE           FDB   TOGGLE
1973 13 67           FDB   SEMIS
                     *
                     * ======>>  115  <<
1975 83              FCB   $83
1976 48              FCC   2,HEX
1978 D8              FCB   $D8
1979 19 61           FDB   SMUDGE-9
197B 15 25     HEX   FDB   DOCOL
197D 10 58           FDB   CLITER
197F 10              FCB   16
1980 16 C8           FDB   BASE,STORE
1984 13 67           FDB   SEMIS
                     *
                     * ======>>  116  <<
1986 87              FCB   $87
1987 44              FCC   6,DECIMAL
198D CC              FCB   $CC
198E 19 75           FDB   HEX-6
1990 15 25     DEC   FDB   DOCOL
1992 10 58           FDB   CLITER
1994 0A              FCB   10          note: hex "A"
1995 16 C8           FDB   BASE,STORE
1999 13 67           FDB   SEMIS
                     *
                     * ######>> screen 42 <<
                     * ======>>  117  <<
199B 87              FCB   $87
```

```
 199C 28              FCC    6,(;CODE)
 19A2 A9              FCB    $A9
 19A3 19 86           FDB    DEC-10
 19A5 15 25     PSCODE FDB   DOCOL,FROMR,TWOP,LATEST,PFA,CFA,STORE
 19B3 13 67           FDB    SEMIS
                *
                * ======>>  118  <<
 19B5 C5              FCB    $C5          immediate
 19B6 3B              FCC    4,;CODE
 19BA C5              FCB    $C5
 19BB 19 9B           FDB    PSCODE-10
 19BD 15 25     SEMIC  FDB   DOCOL,QCSP,COMPIL,PSCODE,SMUDGE,LBRAK,QSTACK
 19CB 13 67           FDB    SEMIS
                * note: "QSTACK" will be replaced by "ASSEMBLER" later
                *
                * ######>> screen 43 <<
                * ======>>  119  <<
 19CD 87              FCB    $87
 19CE 3C              FCC    6,<BUILDS
 19D4 D3              FCB    $D3
 19D5 19 B5           FDB    SEMIC-8
 19D7 15 25     BUILDS FDB   DOCOL,ZERO,CON
 19DD 13 67           FDB    SEMIS
                *
                * ======>>  120  <<
 19DF 85              FCB    $85
 19E0 44              FCC    4,DOES>
 19E4 BE              FCB    $BE
 19E5 19 CD           FDB    BUILDS-10
 19E7 15 25     DOES   FDB   DOCOL,FROMR,TWOP,LATEST,PFA,STORE
 19F3 19 A5           FDB    PSCODE
                *
 19F5 96 F2     DODOES LDA A  IP
 19F7 D6 F3           LDA B  IP+1
 19F9 DE F4           LDX    RP           make room on return stack
 19FB 09              DEX
 19FC 09              DEX
 19FD DF F4           STX    RP
 19FF A7 02           STA A  2,X          push return address
 1A01 E7 03           STA B  3,X
 1A03 DE F0           LDX    W            get addr of pointer to run-time code
 1A05 08              INX
 1A06 08              INX
 1A07 DF E0           STX    N            stash it in scratch area
 1A09 EE 00           LDX    0,X          get new IP
 1A0B DF F2           STX    IP
 1A0D 4F              CLR A               get address of parameter
 1A0E C6 02           LDA B  #2
 1A10 DB E1           ADD B  N+1
 1A12 99 E0           ADC A  N
 1A14 37              PSH B               and push it on data stack
 1A15 36              PSH A
 1A16 7E 10 3A        JMP    NEXT2
                *
                * ######>> screen 44 <<
                * ======>>  121  <<
```

```
1A19 85              FCB   $85
1A1A 43              FCC   4,COUNT
1A1E D4              FCB   $D4
1A1F 19 DF           FDB   DOES-8
1A21 15 25    COUNT  FDB   DOCOL,DUP,ONEP,SWAP,CAT
1A2B 13 67           FDB   SEMIS
                     *
                     * ======>>  122  <<
1A2D 84              FCB   $84
1A2E 54              FCC   3,TYPE
1A31 C5              FCB   $C5
1A32 1A 19           FDB   COUNT-8
1A34 15 25    TYPE   FDB   DOCOL,DDUP,ZBRAN
1A3A 00 18           FDB   TYPE3-*
1A3C 14 5B           FDB   OVER,PLUS,SWAP,XDO
1A44 11 1C    TYPE2  FDB   I,CAT,EMIT,XLOOP
1A4C FF F8           FDB   TYPE2-*
1A4E 10 82           FDB   BRAN
1A50 00 04           FDB   TYPE4-*
1A52 14 6C    TYPE3  FDB   DROP
1A54 13 67    TYPE4  FDB   SEMIS
                     *
                     * ======>>  123  <<
1A56 89              FCB   $89
1A57 2D              FCC   8,-TRAILING
1A5F C7              FCB   $C7
1A60 1A 2D           FDB   TYPE-7
1A62 15 25    DTRAIL FDB   DOCOL,DUP,ZERO,XDO
1A6A 14 5B    DTRAL2 FDB   OVER,OVER,PLUS,ONE,SUB,CAT,BL
1A78 17 69           FDB   SUB,ZBRAN
1A7C 00 08           FDB   DTRAL3-*
1A7E 13 7C           FDB   LEAVE,BRAN
1A82 00 06           FDB   DTRAL4-*
1A84 15 AA    DTRAL3 FDB   ONE,SUB
1A88 10 BA    DTRAL4 FDB   XLOOP
1A8A FF E0           FDB   DTRAL2-*
1A8C 13 67           FDB   SEMIS
                     *
                     * ======>>  124  <<
1A8E 84              FCB   $84
1A8F 28              FCC   3,(.")
1A92 A9              FCB   $A9
1A93 1A 56           FDB   DTRAIL-12
1A95 15 25    PDOTQ  FDB   DOCOL,R,TWOP,COUNT,DUP,ONEP
1AA1 13 A6           FDB   FROMR,PLUS,TOR,TYPE
1AA9 13 67           FDB   SEMIS
                     *
                     * ======>>  125  <<
1AAB C2              FCB   $C2        immediate
1AAC 2E              FCC   1,."
1AAD A2              FCB   $A2
1AAE 1A 8E           FDB   PDOTQ-7
1AB0 15 25    DOTQ   FDB   DOCOL
1AB2 10 58           FDB   CLITER
1AB4 22              FCB   $22        ascii quote
1AB5 16 BD           FDB   STATE,AT,ZBRAN
```

```
1ABB 00 14            FDB    DOTQ1-*
1ABD 19 32            FDB    COMPIL,PDOTQ,WORD
1AC3 17 2C            FDB    HERE,CAT,ONEP,ALLOT,BRAN
1ACD 00 0A            FDB    DOTQ2-*
1ACF 1C 41     DOTQ1  FDB    WORD,HERE,COUNT,TYPE
1AD7 13 67     DOTQ2  FDB    SEMIS
                      *
                      * ######>> screen 45 <<
                      * =======>>  126  <<== MACHINE DEPENDENT
1AD9 86               FCB    $86
1ADA 3F               FCC    5,?STACK
1ADF CB               FCB    $CB
1AE0 1A AB            FDB    DOTQ-5
1AE2 15 25     QSTACK FDB    DOCOL,CLITER
1AE6 12               FCB    $12
1AE7 16 01            FDB    PORIG,AT,TWO,SUB,SPAT,LESS,ONE
1AF5 18 A4            FDB    QERR
                      * prints 'empty stack'
                      *
1AF7 13 37     QSTAC2 FDB    SPAT
                      * Here, we compare with a value at least 128
                      * higher than dict. ptr. (DP)
1AF9 17 2C            FDB    HERE,CLITER
1AFD 80               FCB    $80
1AFE 13 F1            FDB    PLUS,LESS,ZBRAN
1B04 00 06            FDB    QSTAC3-*
1B06 15 B2            FDB    TWO
1B08 18 A4            FDB    QERR
                      * prints 'full stack'
                      *
1B0A 13 67     QSTAC3 FDB    SEMIS
                      *
                      * ======>>  127  <<  this word's function
                      *           is done by ?STACK in this version
                      *       FCB  $85
                      *       FCC 4,?FREE
                      *       FCB $C5
                      *       FDB QSTACK-9
                      * QFREE FDB DOCOL,SPAT,HERE,CLITER
                      *       FCB $80
                      *       FDB PLUS,LESS,TWO,QERR,SEMIS
                      *
                      * ######>> screen 46 <<
                      * ======>>  128  <<
1B0C 86               FCB    $86
1B0D 45               FCC    5,EXPECT
1B12 D4               FCB    $D4
1B13 1A D9            FDB    QSTACK-9
1B15 15 25     EXPECT FDB    DOCOL,OVER,PLUS,OVER,XDO
1B1F 12 39     EXPEC2 FDB    KEY,DUP,CLITER
1B25 0E               FCB    $0E
1B26 16 01            FDB    PORIG,AT,EQUAL,ZBRAN
1B2E 00 1F            FDB    EXPEC3-*
1B30 14 6C            FDB    DROP,CLITER
1B34 08               FCB    8          ( backspace character to emit )
1B35 14 5B            FDB    OVER,I,EQUAL,DUP,FROMR,TWO,SUB,PLUS
```

```
1B45 13 90            FDB   TOR,SUB,BRAN
1B4B 00 27            FDB   EXPEC6-*
1B4D 14 93    EXPEC3  FDB   DUP,CLITER
1B51 0D               FCB   $D          ( carriage return )
1B52 17 75            FDB   EQUAL,ZBRAN
1B56 00 0E            FDB   EXPEC4-*
1B58 13 7C            FDB   LEAVE,DROP,BL,ZERO,BRAN
1B62 00 04            FDB   EXPEC5-*
1B64 14 93    EXPEC4  FDB   DUP
1B66 11 1C    EXPEC5  FDB   I,CSTORE,ZERO,I,ONEP,STORE
1B72 12 21    EXPEC6  FDB   EMIT,XLOOP
1B76 FF A9            FDB   EXPEC2-*
1B78 14 6C            FDB   DROP
1B7A 13 67            FDB   SEMIS
                      *
                      * ======>>  129  <<
1B7C 85               FCB   $85
1B7D 51               FCC   4,QUERY
1B81 D9               FCB   $D9
1B82 1B 0C            FDB   EXPECT-9
1B84 15 25    QUERY   FDB   DOCOL,TIB,AT,COLUMS
1B8C 14 D0            FDB   AT,EXPECT,ZERO,IN,STORE
1B96 13 67            FDB   SEMIS
                      *
                      * ======>>  130  <<
1B98 C1               FCB   $C1         immediate    < carriage return >
1B99 80               FCB   $80
1B9A 1B 7C            FDB   QUERY-8
1B9C 15 25    NULL    FDB   DOCOL,BLK,AT,ZBRAN
1BA4 00 26            FDB   NULL2-*
1BA6 15 AA            FDB   ONE,BLK,PSTORE
1BAC 15 A2            FDB   ZERO,IN,STORE,BLK,AT,BSCR,MOD
1BBA 13 C7            FDB   ZEQU
                      *     check for end of screen
1BBC 10 8E            FDB   ZBRAN
1BBE 00 08            FDB   NULL1-*
1BC0 18 D5            FDB   QEXEC,FROMR,DROP
1BC6 10 82    NULL1   FDB   BRAN
1BC8 00 06            FDB   NULL3-*
1BCA 13 A6    NULL2   FDB   FROMR,DROP
1BCE 13 67    NULL3   FDB   SEMIS
                      *
                      * ######>> screen 47 <<
                      * ======>>  133  <<
1BD0 84               FCB   $84
1BD1 46               FCC   3,FILL
1BD4 CC               FCB   $CC
1BD5 1B 98            FDB   NULL-4
1BD7 15 25    FILL    FDB   DOCOL,SWAP,TOR,OVER,CSTORE,DUP,ONEP
1BE5 13 A6            FDB   FROMR,ONE,SUB,CMOVE
1BED 13 67            FDB   SEMIS
                      *
                      * ======>>  134  <<
1BEF 85               FCB   $85
1BF0 45               FCC   4,ERASE
1BF4 C5               FCB   $C5
```

```
1BF5 1B D0            FDB    FILL-7
1BF7 15 25     ERASE  FDB    DOCOL,ZERO,FILL
1BFD 13 67            FDB    SEMIS
                      *
                      * ======>>  135  <<
1BFF 86               FCB    $86
1C00 42               FCC    5,BLANKS
1C05 D3               FCB    $D3
1C06 1B EF            FDB    ERASE-8
1C08 15 25     BLANKS FDB    DOCOL,BL,FILL
1C0E 13 67            FDB    SEMIS
                      *
                      * ======>>  136  <<
1C10 84               FCB    $84
1C11 48               FCC    3,HOLD
1C14 C4               FCB    $C4
1C15 1B FF            FDB    BLANKS-9
1C17 15 25     HOLD   FDB    DOCOL,LIT,$FFFF,HLD,PSTORE,HLD,AT,CSTORE
1C27 13 67            FDB    SEMIS
                      *
                      * ======>>  137  <<
1C29 83               FCB    $83
1C2A 50               FCC    2,PAD
1C2C C4               FCB    $C4
1C2D 1C 10            FDB    HOLD-7
1C2F 15 25     PAD    FDB    DOCOL,HERE,CLITER
1C35 44               FCB    $44
1C36 13 F1            FDB    PLUS
1C38 13 67            FDB    SEMIS
                      *
                      * ######>> screen 48 <<
                      * ======>>  138  <<
1C3A 84               FCB    $84
1C3B 57               FCC    3,WORD
1C3E C4               FCB    $C4
1C3F 1C 29            FDB    PAD-6
1C41 15 25     WORD   FDB    DOCOL,BLK,AT,ZBRAN
1C49 00 0C            FDB    WORD2-*
1C4B 16 6B            FDB    BLK,AT,BLOCK,BRAN
1C53 00 06            FDB    WORD3-*
1C55 16 23     WORD2  FDB    TIB,AT
1C59 16 74     WORD3  FDB    IN,AT,PLUS,SWAP,ENCLOS,HERE,CLITER
1C67 22               FCB    34
1C68 1C 08            FDB    BLANKS,IN,PSTORE,OVER,SUB,TOR,R,HERE
1C78 14 FF            FDB    CSTORE,PLUS,HERE,ONEP,FROMR,CMOVE
1C84 13 67            FDB    SEMIS
                      *
                      * ######>> screen 49 <<
                      * ======>>  139  <<
1C86 88               FCB    $88
1C87 28               FCC    7,(NUMBER)
1C8E A9               FCB    $A9
1C8F 1C 3A            FDB    WORD-7
1C91 15 25     PNUMB  FDB    DOCOL
1C93 17 10     PNUMB2 FDB    ONEP,DUP,TOR,CAT,BASE,AT,DIGIT,ZBRAN
1CA3 00 2C            FDB    PNUMB4-*
```

```
1CA5 14 7A            FDB    SWAP,BASE,AT,USTAR,DROP,ROT,BASE
1CB3 14 D0            FDB    AT,USTAR,DPLUS,DPL,AT,ONEP,ZBRAN
1CC1 00 08            FDB    PNUMB3-*
1CC3 15 AA            FDB    ONE,DPL,PSTORE
1CC9 13 A6     PNUMB3 FDB    FROMR,BRAN
1CCD FF C6            FDB    PNUMB2-*
1CCF 13 A6     PNUMB4 FDB    FROMR
1CD1 13 67            FDB    SEMIS
                      *
                      * ======>>  140  <<
1CD3 86               FCB    $86
1CD4 4E               FCC    5,NUMBER
1CD9 D2               FCB    $D2
1CDA 1C 86            FDB    PNUMB-11
1CDC 15 25     NUMB   FDB    DOCOL,ZERO,ZERO,ROT,DUP,ONEP,CAT,CLITER
1CEC 2D               FCC    "-"         minus sign
1CED 17 75            FDB    EQUAL,DUP,TOR,PLUS,LIT,$FFFF
1CF9 16 D2     NUMB1  FDB    DPL,STORE,PNUMB,DUP,CAT,BL,SUB
1D07 10 8E            FDB    ZBRAN
1D09 00 15            FDB    NUMB2-*
1D0B 14 93            FDB    DUP,CAT,CLITER
1D11 2E               FCC    "."
1D12 17 69            FDB    SUB,ZERO,QERR,ZERO,BRAN
1D1C FF DD            FDB    NUMB1-*
1D1E 14 6C     NUMB2  FDB    DROP,FROMR,ZBRAN
1D24 00 04            FDB    NUMB3-*
1D26 14 3A            FDB    DMINUS
1D28 13 67     NUMB3  FDB    SEMIS
                      *
                      * ======>>  141  <<
1D2A 85               FCB    $85
1D2B 2D               FCC    4,-FIND
1D2F C4               FCB    $C4
1D30 1C D3            FDB    NUMB-9
1D32 15 25     DFIND  FDB    DOCOL,BL,WORD,HERE,CONTXT,AT,AT
1D40 11 62            FDB    PFIND,DUP,ZEQU,ZBRAN
1D48 00 0A            FDB    DFIND2-*
1D4A 14 6C            FDB    DROP,HERE,LATEST,PFIND
1D52 13 67     DFIND2 FDB    SEMIS
                      *
                      * ######>> screen 50 <<
                      * ======>>  142  <<
1D54 87               FCB    $87
1D55 28               FCC    6,(ABORT)
1D5B A9               FCB    $A9
1D5C 1D 2A            FDB    DFIND-8
1D5E 15 25     PABORT FDB    DOCOL,ABORT
1D62 13 67            FDB    SEMIS
                      *
                      * ======>>  143  <<
1D64 85               FCB    $85
1D65 45               FCC    4,ERROR
1D69 D2               FCB    $D2
1D6A 1D 54            FDB    PABORT-10
1D6C 15 25     ERROR  FDB    DOCOL,WARN,AT,ZLESS
1D74 10 8E            FDB    ZBRAN
```

```
                *  note: WARNING is -1 to abort, 0 to print error #
                *        and 1 to print error message from disc
1D76 00 04            FDB    ERROR2-*
1D78 1D 5E            FDB    PABORT
1D7A 17 2C      ERROR2 FDB    HERE,COUNT,TYPE,PDOTQ
1D82 04               FCB    4,7          ( bell )
1D84 20               FCC    " ? "
1D87 22 7B            FDB    MESS,SPSTOR,IN,AT,BLK,AT,QUIT
1D95 13 67            FDB    SEMIS
                *
                * ======>>  144  <<
1D97 83               FCB    $83
1D98 49               FCC    2,ID.
1D9A AE               FCB    $AE
1D9B 1D 64            FDB    ERROR-8
1D9D 15 25      IDDOT  FDB    DOCOL,PAD,CLITER
1DA3 20               FCB    32
1DA4 10 58            FDB    CLITER
1DA6 5F               FCB    $5F          ( underline )
1DA7 1B D7            FDB    FILL,DUP,PFA,LFA,OVER,SUB,PAD
1DB5 14 7A            FDB    SWAP,CMOVE,PAD,COUNT,CLITER
1DBF 1F               FCB    31
1DC0 13 02            FDB    AND,TYPE,SPACE
1DC6 13 67            FDB    SEMIS
                *
                * ######>> screen 51 <<
                * ======>>  145  <<
1DC8 86               FCB    $86
1DC9 43               FCC    5,CREATE
1DCE C5               FCB    $C5
1DCF 1D 97            FDB    IDDOT-6
1DD1 15 25      CREATE FDB    DOCOL,DFIND,ZBRAN
1DD7 00 1A            FDB    CREAT2-*
1DD9 14 6C            FDB    DROP,PDOTQ
1DDD 08               FCB    8
1DDE 07               FCB    7            ( bel )
1DDF 72               FCC    "redef: "
1DE6 18 68            FDB    NFA,IDDOT,CLITER
1DEC 04               FCB    4
1DED 22 7B            FDB    MESS,SPACE
1DF1 17 2C      CREAT2 FDB    HERE,DUP,CAT,WIDTH,AT,MIN
1DFD 17 10            FDB    ONEP,ALLOT,DUP,CLITER
1E05 A0               FCB    $A0
1E06 14 BE            FDB    TOGGLE,HERE,ONE,SUB,CLITER
1E10 80               FCB    $80
1E11 14 BE            FDB    TOGGLE,LATEST,COMMA,CURENT,AT,STORE
1E1D 17 2C            FDB    HERE,TWOP,COMMA
1E23 13 67            FDB    SEMIS
                *
                * ######>> screen 52 <<
                * ======>>  146  <<
1E25 C9               FCB    $C9          immediate
1E26 5B               FCC    8,[COMPILE]
1E2E DD               FCB    $DD
1E2F 1D C8            FDB    CREATE-9
1E31 15 25      BCOMP  FDB    DOCOL,DFIND,ZEQU,ZERO,QERR,DROP,CFA,COMMA
```

```
1E41 13 67          FDB   SEMIS
                    *
                    * ======>>  147  <<
1E43 C7             FCB   $C7        immediate
1E44 4C             FCC   6,LITERAL
1E4A CC             FCB   $CC
1E4B 1E 25          FDB   BCOMP-12
1E4D 15 25    LITER FDB   DOCOL,STATE,AT,ZBRAN
1E55 00 08          FDB   LITER2-*
1E57 19 32          FDB   COMPIL,LIT,COMMA
1E5D 13 67    LITER2 FDB  SEMIS
                    *
                    * ======>>  148  <<
1E5F C8             FCB   $C8        immediate
1E60 44             FCC   7,DLITERAL
1E67 CC             FCB   $CC
1E68 1E 43          FDB   LITER-10
1E6A 15 25    DLITER FDB  DOCOL,STATE,AT,ZBRAN
1E72 00 08          FDB   DLITE2-*
1E74 14 7A          FDB   SWAP,LITER,LITER
1E7A 13 67    DLITE2 FDB  SEMIS
                    *
                    * ######>> screen 53 <<
                    * ======>>  149  <<
1E7C 89             FCB   $89
1E7D 49             FCC   8,INTERPRET
1E85 D4             FCB   $D4
1E86 1E 5F          FDB   DLITER-11
1E88 15 25    INTERP FDB  DOCOL
1E8A 1D 32    INTER2 FDB  DFIND,ZBRAN
1E8E 00 1C          FDB   INTER5-*
1E90 16 BD          FDB   STATE,AT,LESS
1E96 10 8E          FDB   ZBRAN
1E98 00 0A          FDB   INTER3-*
1E9A 18 5A          FDB   CFA,COMMA,BRAN
1EA0 00 06          FDB   INTER4-*
1EA2 18 5A    INTER3 FDB  CFA,EXEC
1EA6 10 82    INTER4 FDB  BRAN
1EA8 00 1A          FDB   INTER7-*
1EAA 17 2C    INTER5 FDB  HERE,NUMB,DPL,AT,ONEP,ZBRAN
1EB6 00 08          FDB   INTER6-*
1EB8 1E 6A          FDB   DLITER,BRAN
1EBC 00 06          FDB   INTER7-*
1EBE 14 6C    INTER6 FDB  DROP,LITER
1EC2 1A E2    INTER7 FDB  QSTACK,BRAN
1EC6 FF C4          FDB   INTER2-*
                    *     FDB SEMIS never executed
                    *
                    * ######>> screen 54 <<
                    * ======>>  150  <<
1EC8 89             FCB   $89
1EC9 49             FCC   8,IMMEDIATE
1ED1 C5             FCB   $C5
1ED2 1E 7C          FDB   INTERP-12
1ED4 15 25    IMMED FDB   DOCOL,LATEST,CLITER
1EDA 40             FCB   $40
```

```
1EDB 14 BE              FDB    TOGGLE
1EDD 13 67              FDB    SEMIS
                        *
                        * ======>>  151  <<
1EDF 8A                 FCB    $8A
1EE0 56                 FCC    9,VOCABULARY
1EE9 D9                 FCB    $D9
1EEA 1E C8              FDB    IMMED-12
1EEC 15 25      VOCAB   FDB    DOCOL,BUILDS,LIT,$81A0,COMMA,CURENT,AT,CFA
1EFC 17 48              FDB    COMMA,HERE,VOCLIN,AT,COMMA,VOCLIN,STORE,DOES
1F0C 17 1D      DOVOC   FDB    TWOP,CONTXT,STORE
1F12 13 67              FDB    SEMIS
                        *
                        * ======>>  152  <<
                        *
                        * Note: FORTH does not go here in the rom-able dictionary,
                        *     since FORTH is a type of variable.
                        *
                        *
                        * ======>>  153  <<
1F14 8B                 FCB    $8B
1F15 44                 FCC    10,DEFINITIONS
1F1F D3                 FCB    $D3
1F20 1E DF              FDB    VOCAB-13
1F22 15 25      DEFIN   FDB    DOCOL,CONTXT,AT,CURENT,STORE
1F2C 13 67              FDB    SEMIS
                        *
                        * ======>>  154  <<
1F2E C1                 FCB    $C1         immediate  (
1F2F A8                 FCB    $A8
1F30 1F 14              FDB    DEFIN-14
1F32 15 25      PAREN   FDB    DOCOL,CLITER
1F36 29                 FCC    ")"
1F37 1C 41              FDB    WORD
1F39 13 67              FDB    SEMIS
                        *
                        * ######>> screen 55 <<
                        * ======>>  155  <<
1F3B 84                 FCB    $84
1F3C 51                 FCC    3,QUIT
1F3F D4                 FCB    $D4
1F40 1F 2E              FDB    PAREN-4
1F42 15 25      QUIT    FDB    DOCOL,ZERO,BLK,STORE
1F4A 19 48              FDB    LBRAK
                        *
                        *  Here is the outer interpretter
                        *  which gets a line of input, does it, prints " OK"
                        *  then repeats :
                        *
1F4C 13 58      QUIT2   FDB    RPSTOR,CR,QUERY,INTERP,STATE,AT,ZEQU
1F5A 10 8E              FDB    ZBRAN
1F5C 00 08              FDB    QUIT3-*
1F5E 1A 95              FDB    PDOTQ
1F60 03                 FCB    3
1F61 20                 FCC    3, OK
1F64 10 82      QUIT3   FDB    BRAN
```

```
1F66 FF E6              FDB    QUIT2-*
                 *      FDB SEMIS ( never executed )
                 *
                 * ======>>  156  <<
1F68 85                 FCB    $85
1F69 41                 FCC    4,ABORT
1F6D D4                 FCB    $D4
1F6E 1F 3B              FDB    QUIT-7
1F70 15 25       ABORT  FDB    DOCOL,SPSTOR,DEC,QSTACK,DRZERO,CR,PDOTQ
1F7E 08                 FCB    8
1F7F 46                 FCC    "Forth-68"
1F87 01 50              FDB    FORTH,DEFIN
1F8B 1F 42              FDB    QUIT
                 *      FDB SEMIS never executed
```

```
                *
                * ######>> screen 56 <<
                * bootstrap code... moves rom contents to ram :
                * ======>>  157  <<
1F8D 84               FCB    $84
1F8E 43               FCC    3,COLD
1F91 C4               FCB    $C4
1F92 1F 68            FDB    ABORT-8
1F94 1F 96      COLD  FDB    *+2
1F96 8E 01 82   CENT  LDS    #REND-1     top of destination
1F99 CE 20 35         LDX    #ERAM       top of stuff to move
1F9C 09         COLD2 DEX
1F9D A6 00            LDA A  0,X
1F9F 36               PSH A               move TASK & FORTH to ram
1FA0 8C 1F F2         CPX    #RAM
1FA3 26 F7            BNE    COLD2
                *
1FA5 8E 01 0F         LDS    #XFENCE-1   put stack at a safe place fcr now
1FA8 FE 10 22         LDX    COLINT
1FAB FF 01 34         STX    XCOLUM
1FAE FE 10 24         LDX    DELINT
1FB1 FF 01 32         STX    XDELAY
1FB4 FE 10 20         LDX    VOCINT
1FB7 FF 01 14         STX    XVOCL
1FBA FE 10 1E         LDX    DPINIT
1FBD FF 01 12         STX    XDP
1FC0 FE 10 1C         LDX    FENCIN
1FC3 FF 01 10         STX    XFENCE


1FC6 8E 01 0F   WENT  LDS    #XFENCE-1   top of destination
1FC9 CE 10 1C         LDX    #FENCIN     top of stuff to move
1FCC 09         WARM2 DEX
1FCD A6 00            LDA A  0,X
1FCF 36               PSH A
1FD0 8C 10 12         CPX    #SINIT
1FD3 26 F7            BNE    WARM2
                *
1FD5 BE 10 12         LDS    SINIT
1FD8 FE 10 10         LDX    UPINIT
1FDB DF F6            STX    UP          init user ram pointer
1FDD CE 1F 70         LDX    #ABORT
1FE0 DF F2            STX    IP
1FE2 01               NOP                Here is a place to jump to special user
1FE3 01               NOP                initializations such as I/O interrups
1FE4 01               NOP
                *
                *    For systems with TRACE:
1FE5 CE 00 00         LDX    #00
1FE8 DF EA            STX    TRLIM       clear trace mode
1FEA CE 00 00         LDX    #0
1FED DF EC            STX    BRKPT       clear breakpoint address
1FEF 7E 13 5A         JMP    RPSTOR+2    start the virtual machine running !
```

```
                    *
                    * Here is the stuff that gets copied to ram :
                    * at address $140:
                    *
1FF2 30 00          RAM     FDB     $3000,$3000,0,0

                    * ======>> (152) <<
1FFA C5                     FCB     $C5         immediate
1FFB 46                     FCC     4,FORTH
1FFF C8                     FCB     $C8
2000 27 40                  FDB     NOOP-7
2002 19 F5          RFORTH  FDB     DODOES,DOVOC,$81A0,TASK-7
200A 00 00                  FDB     0
200C 28                     FCC     "(C) Forth Interest Group, 1979"
202A 84                     FCB     $84
202B 54                     FCC     3,TASK
202E CB                     FCB     $CB
202F 01 48                  FDB     FORTH-8
2031 15 25          RTASK   FDB     DOCOL,SEMIS
2035 44             ERAM    FCC     "David Lion"
```

```
                *
                * ######>> screen 57 <<
                * ======>>  158  <<
203F 84               FCB   $84
2040 53               FCC   3,S->D
2043 C4               FCB   $C4
2044 1F 8D            FDB   COLD-7
2046 15 25      STOD  FDB   DOCOL,DUP,ZLESS,MINUS
204E 13 67            FDB   SEMIS



                *
                * ======>>  159  <<
2050 81               FCB   $81       *
2051 AA               FCB   $AA
2052 20 3F            FDB   STOD-7
2054 20 56      STAR  FDB   *+2
2056 BD 12 AB         JSR   USTARS
2059 31               INS
205A 31               INS
205B 7E 10 34         JMP   NEXT
                *
                * ======>>  160  <<
205E 84               FCB   $84
205F 2F               FCC   3,/MOD
2062 C4               FCB   $C4
2063 20 50            FDB   STAR-4
2065 15 25      SLMOD FDB   DOCOL,TOR,STOD,FROMR,USLASH
206F 13 67            FDB   SEMIS
                *
                * ======>>  161  <<
2071 81               FCB   $81       /
2072 AF               FCB   $AF
2073 20 5E            FDB   SLMOD-7
2075 15 25      SLASH FDB   DOCOL,SLMOD,SWAP,DROP
207D 13 67            FDB   SEMIS
                *
                * ======>>  162  <<
207F 83               FCB   $83
2080 4D               FCC   2,MOD
2082 C4               FCB   $C4
2083 20 71            FDB   SLASH-4
2085 15 25      MOD   FDB   DOCOL,SLMOD,DROP
208B 13 67            FDB   SEMIS
                *
                * ======>>  163  <<
208D 85               FCB   $85
208E 2A               FCC   4,*/MOD
2092 C4               FCB   $C4
2093 20 7F            FDB   MOD-6
2095 15 25      SSMOD FDB   DOCOL,TOR,USTAR,FROMR,USLASH
209F 13 67            FDB   SEMIS
                *
                * ======>>  164  <<
```

```
20A1 82              FCB   $82
20A2 2A              FCC   1,*/
20A3 AF              FCB   $AF
20A4 20 8D           FDB   SSMOD-8
20A6 15 25     SSLASH FDB  DOCOL,SSMOD,SWAP,DROP
20AE 13 67           FDB   SEMIS
                     *
                     * ======>>  165  <<
20B0 85              FCB   $85
20B1 4D              FCC   4,M/MOD
20B5 C4              FCB   $C4
20B6 20 A1           FDB   SSLASH-5
20B8 15 25     MSMOD FDB   DOCOL,TOR,ZERO,R,USLASH
20C2 13 A6           FDB   FROMR,SWAP,TOR,USLASH,FROMR
20CC 13 67           FDB   SEMIS
                     *
                     * ======>>  166  <<
20CE 83              FCB   $83
20CF 41              FCC   2,ABS
20D1 D3              FCB   $D3
20D2 20 B0           FDB   MSMOD-8
20D4 15 25     ABS   FDB   DOCOL,DUP,ZLESS,ZBRAN
20DC 00 04           FDB   ABS2-*
20DE 14 21           FDB   MINUS
20E0 13 67     ABS2  FDB   SEMIS
                     *
                     * ======>>  167  <<
20E2 84              FCB   $84
20E3 44              FCC   3,DABS
20E6 D3              FCB   $D3
20E7 20 CE           FDB   ABS-6
20E9 15 25     DABS  FDB   DOCOL,DUP,ZLESS,ZBRAN
20F1 00 04           FDB   DABS2-*
20F3 14 3A           FDB   DMINUS
20F5 13 67     DABS2 FDB   SEMIS
                     *
                     * ######>> screen 58 <<
                     * Disc primatives :
                     * ======>>  168  <<
20F7 83              FCB   $83
20F8 55              FCC   2,USE
20FA C5              FCB   $C5
20FB 20 E2           FDB   DABS-7
20FD 15 5F     USE   FDB   DOCON
20FF 01 40           FDB   XUSE
                     *
                     * ======>>  169  <<
2101 84              FCB   $84
2102 50              FCC   3,PREV
2105 D6              FCB   $D6
2106 20 F7           FDB   USE-6
2108 15 5F     PREV  FDB   DOCON
210A 01 42           FDB   XPREV
                     *
                     * ======>>  170  <<
210C 84              FCB   $84
```

```
210D 2B              FCC   3,+BUF
2110 C6              FCB   $C6
2111 21 01           FDB   PREV-7
2113 15 25     PBUF  FDB   DOCOL,CLITER
2117 84              FCB   $84
2118 13 F1           FDB   PLUS,DUP,LIMIT,EQUAL,ZBRAN
2122 00 06           FDB   PBUF2-*
2124 14 6C           FDB   DROP,FIRST
2128 14 93     PBUF2 FDB   DUP,PREV,AT,SUB
2130 13 67           FDB   SEMIS
                     *
                     * ======>>  171  <<
2132 86              FCB   $86
2133 55              FCC   5,UPDATE
2138 C5              FCB   $C5
2139 21 0C           FDB   PBUF-7
213B 15 25     UPDATE FDB  DOCOL,PREV,AT,AT,LIT,$8000,OR,PREV,AT,STORE
214F 13 67           FDB   SEMIS
                     *
                     * ======>>  172  <<
2151 8D              FCB   $8D
2152 45              FCC   12,EMPTY-BUFFERS
215E D3              FCB   $D3
215F 21 32           FDB   UPDATE-9
2161 15 25     MTBUF FDB   DOCOL,FIRST,LIMIT,OVER,SUB,ERASE
216D 13 67           FDB   SEMIS
                     *
                     * ======>>  173  <<
216F 83              FCB   $83
2170 44              FCC   2,DR0
2172 B0              FCB   $B0
2173 21 51           FDB   MTBUF-16
2175 15 25     DRZERO FDB  DOCOL,ZERO,OFSET,STORE
217D 13 67           FDB   SEMIS
                     *
                     * ======>>  174  <<== system dependant word
217F 83              FCB   $83
2180 44              FCC   2,DR1
2182 B1              FCB   $B1
2183 21 6F           FDB   DRZERO-6
2185 15 25     DRONE FDB   DOCOL,LIT,$07D0,OFSET,STORE
218F 13 67           FDB   SEMIS
                     *
                     * ######>> screen 59 <<
                     * ======>>  175  <<
2191 86              FCB   $86
2192 42              FCC   5,BUFFER
2197 D2              FCB   $D2
2198 21 7F           FDB   DRONE-6
219A 15 25     BUFFER FDB  DOCOL,USE,AT,DUP,TOR
21A4 21 13     BUFFR2 FDB  PBUF,ZBRAN
21A8 FF FC           FDB   BUFFR2-*
21AA 20 FD           FDB   USE,STORE,R,AT,ZLESS
21B4 10 8E           FDB   ZBRAN
21B6 00 14           FDB   BUFFR3-*
21B8 13 B9           FDB   R,TWOP,R,AT,LIT,$7FFF,AND,ZERO,RW
```

```
21CA 13 B9      BUFFR3 FDB    R,STORE,R,PREV,STORE,FROMR,TWOP
21D8 13 67             FDB    SEMIS
                *
                * ######>> screen 60 <<
                * ======>>  176  <<
21DA 85                FCB    $85
21DB 42                FCC    4,BLOCK
21DF CB                FCB    $CB
21E0 21 91             FDB    BUFFER-9
21E2 15 25      BLOCK  FDB    DOCOL,OFSET,AT,PLUS,TOR
21EC 21 08             FDB    PREV,AT,DUP,AT,R,SUB,DUP,PLUS,ZBRAN
21FE 00 34             FDB    BLOCK5-*
2200 21 13      BLOCK3 FDB    PBUF,ZEQU,ZBRAN
2206 00 14             FDB    BLOCK4-*
2208 14 6C             FDB    DROP,R,BUFFER,DUP,R,ONE,RW,TWO,SUB
221A 14 93      BLOCK4 FDB    DUP,AT,R,SUB,DUP,PLUS,ZEQU,ZBRAN
222A FF D6             FDB    BLOCK3-*
222C 14 93             FDB    DUP,PREV,STORE
2232 13 A6      BLOCK5 FDB    FROMR,DROP,TWOP
2238 13 67             FDB    SEMIS
                *
                * ######>> screen 61 <<
                * ======>>  177  <<
223A 86                FCB    $86
223B 28                FCC    5,(LINE)
2240 A9                FCB    $A9
2241 21 DA             FDB    BLOCK-8
2243 15 25      PLINE  FDB    DOCOL,TOR,CLITER
2249 40                FCB    $40
224A 15 E7             FDB    BBUF,SSMOD,FROMR,BSCR,STAR,PLUS,BLOCK,PLUS,CLITER
225C 40                FCB    $40
225D 13 67             FDB    SEMIS
                *
                * ======>>  178  <<
225F 85                FCB    $85
2260 2E                FCC    4,.LINE
2264 C5                FCB    $C5
2265 22 3A             FDB    PLINE-9
2267 15 25      DLINE  FDB    DOCOL,PLINE,DTRAIL,TYPE
226F 13 67             FDB    SEMIS
                *
                * ======>>  179  <<
2271 87                FCB    $87
2272 4D                FCC    6,MESSAGE
2278 C5                FCB    $C5
2279 22 5F             FDB    DLINE-8
227B 15 25      MESS   FDB    DOCOL,WARN,AT,ZBRAN
2283 00 1B             FDB    MESS3-*
2285 18 00             FDB    DDUP,ZBRAN
2289 00 15             FDB    MESS3-*
228B 10 58             FDB    CLITER
228D 04                FCB    4
228E 16 95             FDB    OFSET,AT,BSCR,SLASH,SUB,DLINE,BRAN
229C 00 0D             FDB    MESS4-*
229E 1A 95      MESS3  FDB    PDOTQ
22A0 06                FCB    6
```

```
22A1 65              FCC    6,err #
22A7 26 34           FDB    DOT
22A9 13 67     MESS4 FDB    SEMIS
               *
               * ######>> screen 62 <<
               * ======>>  180  <<
22AB 84              FCB    $84
22AC 4C              FCC    3,LOAD input:scr #
22AF C4              FCB    $C4
22B0 22 71           FDB    MESS-10
22B2 15 25     LOAD  FDB    DOCOL,BLK,AT,TOR,IN,AT,TOR,ZERO,IN,STORE
22C6 15 F3           FDB    BSCR,STAR,BLK,STORE
22CE 1E 88           FDB    INTERP,FROMR,IN,STORE,FROMR,BLK,STORE
22DC 13 67           FDB    SEMIS
               *
               * ======>>  181  <<
22DE C3              FCB    $C3
22DF 2D              FCC    2,-->
22E1 BE              FCB    $BE
22E2 22 AB           FDB    LOAD-7
22E4 15 25     ARROW FDB    DOCOL,QLOAD,ZERO,IN,STORE,BSCR
22F0 16 6B           FDB    BLK,AT,OVER,MOD,SUB,BLK,PSTORE
22FE 13 67           FDB    SEMIS
```

```
                *
                * ######>> screen 63 <<
                *    The next 4 subroutines are machine dependant, and are
                *    called by words 13 through 16 in the dictionary.
                *
                * ======>>  182  << code for EMIT
2300 D7 E0      PEMIT  STA B  N          save B
2302 DF E1             STX    N+1        save X
2304 F6 FB CE          LDA B  ACIAC
2307 C5 02             BIT B  #2         check ready bit
2309 27 F9             BEQ    PEMIT+4    if not ready for more data
230B B7 FB CF          STA A  ACIAD
230E DE F6             LDX    UP
2310 E7 36             STA B  IOSTAT-UORIG,X
2312 D6 E0             LDA B  N          recover B & X
2314 DE E1             LDX    N+1
2316 39                RTS               only A register may change
                *  PEMIT JMP $E1D1     for MIKBUG
                *  PEMIT FCB $3F,$11,$39 for PROTO
                *  PEMIT JMP $D286   for Smoke Signal DOS
                *
                * ======>>  183  << code for KEY
2317 D7 E0      PKEY   STA B  N
2319 DF E1             STX    N+1
231B F6 FB CE          LDA B  ACIAC
231E 57                ASR B
231F 24 FA             BCC    PKEY+4     no incomming data yet
2321 B6 FB CF          LDA A  ACIAD
2324 84 7F             AND A  #$7F       strip parity bit
2326 DE F6             LDX    UP
2328 E7 37             STA B  IOSTAT+1-UORIG,X
232A D6 E0             LDA B  N
232C DE E1             LDX    N+1
232E 39                RTS
                *  PKEY JMP $E1AC     for MIKBUG
                *  PKEY FCB $3F,$14,$39 for AMI PROTO
                *  PKEY JMP $D289   for Smoke Signal DOS
                *
                * ######>> screen 64 <<
                * ======>>  184  << code for ?TERMINAL
232F B6 FB CE   PQTER  LDA A  ACIAC      Test for 'break'  condition
2332 84 11             AND A  #$11       mask framing error bit and
                *        input buffer full
2334 27 05             BEQ    PQTER2
2336 B6 FB CF          LDA A  ACIAD      clear input buffer
2339 86 01             LDA A  #01
233B 39         PQTER2 RTS
```

```
                *
                * ======>>  185  << code for CR
233C 86 0D      PCR     LDA A  #$D        carriage return
233E 8D C0              BSR    PEMIT
2340 86 0A              LDA A  #$A        line feed
2342 8D BC              BSR    PEMIT
2344 86 7F              LDA A  #$7F       rubout
2346 DE F6              LDX    UP
2348 E6 33              LDA B  XDELAY+1-UORIG,X
234A 5A         PCR2    DEC B
234B 2B EE              BMI    PQTER2     return if minus
234D 37                 PSH B             save counter
234E 8D B0              BSR    PEMIT      print RUBOUTs to delay.....
2350 33                 PUL B
2351 20 F7              BRA    PCR2       repeat
```

```
                *
                * ######>> screen 66 <<
                * ======>>  187  <<
2353 85               FCB    $85
2354 3F               FCC    4,?DISC
2358 C3               FCB    $C3
2359 22 DE            FDB    ARROW-6
235B 23 5D      QDISC FDB    *+2
235D 7E 10 34         JMP    NEXT
                *
                * ######>> screen 67 <<
                * ======>>  189  <<
2360 8B               FCB    $8B
2361 42               FCC    10,BLOCK-WRITE
236B C5               FCB    $C5
236C 23 53            FDB    QDISC-8
236E 23 70      BWRITE FDB   *+2
2370 7E 10 34         JMP    NEXT
                *
                * ######>> screen 68 <<
                * ======>>  190  <<
2373 8A               FCB    $8A
2374 42               FCC    9,BLOCK-READ
237D C4               FCB    $C4
237E 23 60            FDB    BWRITE-14
2380 23 82      BREAD  FDB   *+2
2382 7E 10 34         JMP    NEXT
                *
                *The next 3 words are written to create a substitute for disc
                * mass memory,located between $3210 & $3FFF in ram.
                * ======>>  190.1  <<
2385 82               FCB    $82
2386 4C               FCC    1,LO
2387 CF               FCB    $CF
2388 23 73            FDB    BREAD-13
238A 15 5F      LO     FDB    DOCON
238C 32 10            FDB    MEMEND     a system dependant equate at front
                *
                * ======>>  190.2  <<
238E 82               FCB    $82
238F 48               FCC    1,HI
2390 C9               FCB    $C9
2391 23 85            FDB    LO-5
2393 15 5F      HI     FDB    DOCON
2395 3F FF            FDB    MEMTOP     ( $3FFF in this version )
                *
                * ######>> screen 69 <<
                * ======>>  191  <<
2397 83               FCB    $83
2398 52               FCC    2,R/W
239A D7               FCB    $D7
239B 23 8E            FDB    HI-5
239D 15 25      RW     FDB    DOCOL,TOR,BBUF,STAR,LO,PLUS,DUP,HI,GREAT,ZBRAN
23B1 00 0F            FDB    RW2-*
23B3 1A 95            FDB    PDOTQ
```

```
23B5 08              FCB   8
23B6 20              FCC   8, Range ?
23BE 1F 42           FDB   QUIT
23C0 13 A6     RW2   FDB   FROMR,ZBRAN
23C4 00 04           FDB   RW3-*
23C6 14 7A           FDB   SWAP
23C8 15 E7     RW3   FDB   BBUF,CMOVE
23CC 13 67           FDB   SEMIS
                     *
                     * ######>> screen 72 <<
                     * ======>>  192  <<
23CE C1              FCB   $C1        immediate
23CF A7              FCB   $A7        '  ( tick )
23D0 23 97           FDB   RW-6
23D2 15 25     TICK  FDB   DOCOL,DFIND,ZEQU,ZERO,QERR,DROP,LITER
23E0 13 67           FDB   SEMIS
                     *
                     * ======>>  193  <<
23E2 86              FCB   $86
23E3 46              FCC   5,FORGET
23E8 D4              FCB   $D4
23E9 23 CE           FDB   TICK-4
23EB 15 25     FORGET FDB  DOCOL,CURENT,AT,CONTXT,AT,SUB,CLITER
23F9 18              FCB   $18
23FA 18 A4           FDB   QERR,TICK,DUP,FENCE,AT,LESS,CLITER
2408 15              FCB   $15
2409 18 A4           FDB   QERR,DUP,ZERO,PORIG,GREAT,CLITER
2415 15              FCB   $15
2416 18 A4           FDB   QERR,DUP,NFA,DP,STORE,LFA,AT,CONTXT,AT,STORE
242A 13 67           FDB   SEMIS
                     *
                     * ######>> screen 73 <<
                     * ======>>  194  <<
242C 84              FCB   $84
242D 42              FCC   3,BACK
2430 CB              FCB   $CB
2431 23 E2           FDB   FORGET-9
2433 15 25     BACK  FDB   DOCOL,HERE,SUB,COMMA
243B 13 67           FDB   SEMIS
                     *
                     * ======>>  195  <<
243D C5              FCB   $C5
243E 42              FCC   4,BEGIN
2442 CE              FCB   $CE
2443 24 2C           FDB   BACK-7
2445 15 25     BEGIN FDB   DOCOL,QCOMP,HERE,ONE
244D 13 67           FDB   SEMIS
                     *
                     * ======>>  196  <<
244F C5              FCB   $C5
2450 45              FCC   4,ENDIF
2454 C6              FCB   $C6
2455 24 3D           FDB   BEGIN-8
2457 15 25     ENDIF FDB   DOCOL,QCOMP,TWO,QPAIRS,HERE
2461 14 5B           FDB   OVER,SUB,SWAP,STORE
2469 13 67           FDB   SEMIS
```

```
                    *
                    * ======>>  197  <<
246B C4                     FCB    $C4
246C 54                     FCC    3,THEN
246F CE                     FCB    $CE
2470 24 4F                  FDB    ENDIF-8
2472 15 25          THEN    FDB    DOCOL,ENDIF
2476 13 67                  FDB    SEMIS
                    *
                    * ======>>  198  <<
2478 C2                     FCB    $C2
2479 44                     FCC    1,DO
247A CF                     FCB    $CF
247B 24 6B                  FDB    THEN-7
247D 15 25          DO      FDB    DOCOL,COMPIL,XDO,HERE,THREE
2487 13 67                  FDB    SEMIS
                    *
                    * ======>>  199  <<
2489 C4                     FCB    $C4
248A 4C                     FCC    3,LOOP
248D D0                     FCB    $D0
248E 24 78                  FDB    DO-5
2490 15 25          LOOP    FDB    DOCOL,THREE,QPAIRS,COMPIL,XLOOP,BACK
249C 13 67                  FDB    SEMIS
                    *
                    * ======>>  200  <<
249E C5                     FCB    $C5
249F 2B                     FCC    4,+LOOP
24A3 D0                     FCB    $D0
24A4 24 89                  FDB    LOOP-7
24A6 15 25          PLOOP   FDB    DOCOL,THREE,QPAIRS,COMPIL,XPLOOP,BACK
24B2 13 67                  FDB    SEMIS
                    *
                    * ======>>  201  <<
24B4 C5                     FCB    $C5
24B5 55                     FCC    4,UNTIL     ( same as END )
24B9 CC                     FCB    $CC
24BA 24 9E                  FDB    PLOOP-8
24BC 15 25          UNTIL   FDB    DOCOL,ONE,QPAIRS,COMPIL,ZBRAN,BACK
24C8 13 67                  FDB    SEMIS
                    *
                    * ######>> screen 74 <<
                    * ======>>  202  <<
24CA C3                     FCB    $C3
24CB 45                     FCC    2,END
24CD C4                     FCB    $C4
24CE 24 B4                  FDB    UNTIL-8
24D0 15 25          END     FDB    DOCOL,UNTIL
24D4 13 67                  FDB    SEMIS
                    *
                    * ======>>  203  <<
24D6 C5                     FCB    $C5
24D7 41                     FCC    4,AGAIN
24DB CE                     FCB    $CE
24DC 24 CA                  FDB    END-6
24DE 15 25          AGAIN   FDB    DOCOL,ONE,QPAIRS,COMPIL,BRAN,BACK
```

```
24EA 13 67            FDB    SEMIS
                      *
                      * ======>>  204  <<
24EC C6               FCB    $C6
24ED 52               FCC    5,REPEAT
24F2 D4               FCB    $D4
24F3 24 D6            FDB    AGAIN-8
24F5 15 25      REPEAT FDB   DOCOL,TOR,TOR,AGAIN,FROMR,FROMR
2501 15 B2            FDB    TWO,SUB,ENDIF
2507 13 67            FDB    SEMIS
                      *
                      * ======>>  205  <<
2509 C2               FCB    $C2
250A 49               FCC    1,IF
250B C6               FCB    $C6
250C 24 EC            FDB    REPEAT-9
250E 15 25      IF    FDB    DOCOL,COMPIL,ZBRAN,HERE,ZERO,COMMA,TWO
251C 13 67            FDB    SEMIS
                      *
                      * ======>>  206  <<
251E C4               FCB    $C4
251F 45               FCC    3,ELSE
2522 C5               FCB    $C5
2523 25 09            FDB    IF-5
2525 15 25      ELSE  FDB    DOCOL,TWO,QPAIRS,COMPIL,BRAN,HERE
2531 15 A2            FDB    ZERO,COMMA,SWAP,TWO,ENDIF,TWO
253D 13 67            FDB    SEMIS
                      *
                      * ======>>  207  <<
253F C5               FCB    $C5
2540 57               FCC    4,WHILE
2544 C5               FCB    $C5
2545 25 1E            FDB    ELSE-7
2547 15 25      WHILE FDB    DOCOL,IF,TWOP
254D 13 67            FDB    SEMIS
                      *
                      * ######>> screen 75 <<
                      * ======>>  208  <<
254F 86               FCB    $86
2550 53               FCC    5,SPACES
2555 D3               FCB    $D3
2556 25 3F            FDB    WHILE-8
2558 15 25      SPACES FDB   DOCOL,ZERO,MAX,DDUP,ZBRAN
2562 00 0C            FDB    SPACE3-*
2564 15 A2            FDB    ZERO,XDO
2568 17 C1      SPACE2 FDB   SPACE,XLOOP
256C FF FC            FDB    SPACE2-*
256E 13 67      SPACE3 FDB   SEMIS
                      *
                      * ======>>  209  <<
2570 82               FCB    $82
2571 3C               FCC    1,<#
2572 A3               FCB    $A3
2573 25 4F            FDB    SPACES-9
2575 15 25      BDIGS  FDB   DOCOL,PAD,HLD,STORE
257D 13 67            FDB    SEMIS
```

```
                    *
                    * ======>>  210  <<
257F 82                 FCB   $82
2580 23                 FCC   1,#>
2581 BE                 FCB   $BE
2582 25 70              FDB   BDIGS-5
2584 15 25      EDIGS   FDB   DOCOL,DROP,DROP,HLD,AT,PAD,OVER,SUB
2594 13 67              FDB   SEMIS
                    *
                    * ======>>  211  <<
2596 84                 FCB   $84
2597 53                 FCC   3,SIGN
259A CE                 FCB   $CE
259B 25 7F              FDB   EDIGS-5
259D 15 25      SIGN    FDB   DOCOL,ROT,ZLESS,ZBRAN
25A5 00 07              FDB   SIGN2-*
25A7 10 58              FDB   CLITER
25A9 2D                 FCC   "-"
25AA 1C 17              FDB   HOLD
25AC 13 67      SIGN2   FDB   SEMIS
                    *
                    * ======>>  212  <<
25AE 81                 FCB   $81         #
25AF A3                 FCB   $A3
25B0 25 96              FDB   SIGN-7
25B2 15 25      DIG     FDB   DOCOL,BASE,AT,MSMOD,ROT,CLITER
25BE 09                 FCB   9
25BF 14 5B              FDB   OVER,LESS,ZBRAN
25C5 00 07              FDB   DIG2-*
25C7 10 58              FDB   CLITER
25C9 07                 FCB   7
25CA 13 F1              FDB   PLUS
25CC 10 58      DIG2    FDB   CLITER
25CE 30                 FCC   "0"         ascii zero
25CF 13 F1              FDB   PLUS,HOLD
25D3 13 67              FDB   SEMIS
                    *
                    * ======>>  213  <<
25D5 82                 FCB   $82
25D6 23                 FCC   1,#S
25D7 D3                 FCB   $D3
25D8 25 AE              FDB   DIG-4
25DA 15 25      DIGS    FDB   DOCOL
25DC 25 B2      DIGS2   FDB   DIG,OVER,OVER,OR,ZEQU,ZBRAN
25E8 FF F4              FDB   DIGS2-*
25EA 13 67              FDB   SEMIS
                    *
                    * ######>> screen 76 <<
                    * ======>>  214  <<
25EC 82                 FCB   $82
25ED 2E                 FCC   1,.R
25EE D2                 FCB   $D2
25EF 25 D5              FDB   DIGS-5
25F1 15 25      DOTR    FDB   DOCOL,TOR,STOD,FROMR,DDOTR
25FB 13 67              FDB   SEMIS
                    *
```

```
                * ======>>  215  <<
25FD 83                 FCB    $83
25FE 44                 FCC    2,D.R
2600 D2                 FCB    $D2
2601 25 EC              FDB    DOTR-5
2603 15 25      DDOTR   FDB    DOCOL,TOR,SWAP,OVER,DABS,BDIGS,DIGS,SIGN
2613 25 84              FDB    EDIGS,FROMR,OVER,SUB,SPACES,TYPE
261F 13 67              FDB    SEMIS
                *
                * ======>>  216  <<
2621 82                 FCB    $82
2622 44                 FCC    1,D.
2623 AE                 FCB    $AE
2624 25 FD              FDB    DDOTR-6
2626 15 25      DDOT    FDB    DOCOL,ZERO,DDOTR,SPACE
262E 13 67              FDB    SEMIS
                *
                * ======>>  217  <<
2630 81                 FCB    $81        .
2631 AE                 FCB    $AE
2632 26 21              FDB    DDOT-5
2634 15 25      DOT     FDB    DOCOL,STOD,DDOT
263A 13 67              FDB    SEMIS
                *
                * ======>>  218  <<
263C 81                 FCB    $81        ?
263D BF                 FCB    $BF
263E 26 30              FDB    DOT-4
2640 15 25      QUEST   FDB    DOCOL,AT,DOT
2646 13 67              FDB    SEMIS
                *
                * ######>> screen 77 <<
                * ======>>  219  <<
2648 84                 FCB    $84
2649 4C                 FCC    3,LIST
264C D4                 FCB    $D4
264D 26 3C              FDB    QUEST-4
264F 15 25      LIST    FDB    DOCOL,DEC,CR,DUP,SCR,STORE,PDOTQ
265D 06                 FCB    6
265E 53                 FCC    "SCR # "
2664 26 34              FDB    DOT,CLITER
2668 10                 FCB    $10
2669 15 A2              FDB    ZERO,XDO
266D 12 5E      LIST2   FDB    CR,I,THREE
2673 25 F1              FDB    DOTR,SPACE,I,SCR,AT,DLINE,XLOOP
2681 FF EC              FDB    LIST2-*
2683 12 5E              FDB    CR
2685 13 67              FDB    SEMIS
                *
                * ======>>  220  <<
2687 85                 FCB    $85
2688 49                 FCC    4,INDEX
268C D8                 FCB    $D8
268D 26 48              FDB    LIST-7
268F 15 25      INDEX   FDB    DOCOL,CR,ONEP,SWAP,XDO
2699 12 5E      INDEX2  FDB    CR,I,THREE
```

```
269F 25 F1            FDB    DOTR,SPACE,ZERO,I,DLINE
26A9 12 50            FDB    QTERM,ZBRAN
26AD 00 04            FDB    INDEX3-*
26AF 13 7C            FDB    LEAVE
26B1 10 BA    INDEX3  FDB    XLOOP
26B3 FF E6            FDB    INDEX2-*
26B5 13 67            FDB    SEMIS
                      *
                      * ======>>  221  <<
26B7 85               FCB    $85
26B8 54               FCC    4,TRIAD
26BC C4               FCB    $C4
26BD 26 87            FDB    INDEX-8
26BF 15 25    TRIAD   FDB    DOCOL,THREE,SLASH,THREE,STAR
26C9 15 BA            FDB    THREE,OVER,PLUS,SWAP,XDO
26D3 12 5E    TRIAD2  FDB    CR,I
26D7 26 4F            FDB    LIST,QTERM,ZBRAN
26DD 00 04            FDB    TRIAD3-*
26DF 13 7C            FDB    LEAVE
26E1 10 BA    TRIAD3  FDB    XLOOP
26E3 FF F0            FDB    TRIAD2-*
26E5 12 5E            FDB    CR,CLITER
26E9 0F               FCB    $0F
26EA 22 7B            FDB    MESS,CR
26EE 13 67            FDB    SEMIS
                      *
                      * ######>> screen 78 <<
                      * ======>>  222  <<
26F0 85               FCB    $85
26F1 56               FCC    4,VLIST
26F5 D4               FCB    $D4
26F6 26 B7            FDB    TRIAD-8
26F8 15 25    VLIST   FDB    DOCOL,CLITER
26FC 80               FCB    $80
26FD 16 7E            FDB    OUT,STORE,CONTXT,AT,AT
2707 16 7E    VLIST1  FDB    OUT,AT,COLUMS,AT,CLITER
2711 20               FCB    32
2712 17 69            FDB    SUB,GREAT,ZBRAN
2718 00 0A            FDB    VLIST2-*
271A 12 5E            FDB    CR,ZERO,OUT,STORE
2722 14 93    VLIST2  FDB    DUP,IDDOT,SPACE,SPACE,PFA,LFA,AT
2730 14 93            FDB    DUP,ZEQU,QTERM,OR,ZBRAN
273A FF CD            FDB    VLIST1-*
273C 14 6C            FDB    DROP
273E 13 67            FDB    SEMIS
                      *
                      * ======>>  XX  <<
2740 84               FCB    $84
2741 4E               FCC    3,NOOP
2744 D0               FCB    $D0
2745 26 F0            FDB    VLIST-8
2747 10 34    NOOP    FDB    NEXT        a useful no-op
2749 00 00    ZZZZ    FDB    0,0,0,0,0,0,0,0 end of rom program
```

END
NO ERROR(S) DETECTED

SYMBOL TABLE:

ABORT	1F70	ABS	20D4	ABS2	20E0	ACIAC	FBCE
ACIAD	FBCF	AGAIN	24DE	ALLOT	173C	AND	1302
ARROW	22E4	AT	14D0	BACK	2433	BACKSP	100E
BASE	16C8	BBUF	15E7	BCOMP	1E31	BDIGS	2575
BEGIN	2445	BL	15C3	BLANK	2775	BLANKS	1C08
BLK	166B	BLOCK	21E2	BLOCK3	2200	BLOCK4	221A
BLOCK5	2232	BRAN	1082	BREAD	2380	BRKPT	00EC
BSCR	15F3	BUFFER	219A	BUFFR2	21A4	BUFFR3	21CA
BUILDS	19D7	BWRITE	236E	CAT	14DF	CCOMM	1759
CENT	1F96	CFA	185A	CLITER	1058	CMOV1	1275
CMOV2	127C	CMOV3	129A	CMOVE	126E	CNT	280F
COLD	1F94	COLD2	1F9C	COLINT	1022	COLON	1511
COLUMS	1707	COMMA	1748	COMPIL	1932	CON	1555
CONHB	3F15	CONTXT	16A3	COUNT	1A21	CR	125E
CREAT2	1DF1	CREATE	1DD1	CSP	16E6	CSTORE	14FF
CURENT	16B1	DABS	20E9	DABS2	20F5	DDOT	2626
DDOTR	2603	DDUP	1800	DDUP2	180A	DEC	1990
DEFIN	1F22	DELINT	1024	DFIND	1D32	DFIND2	1D52
DIG	25B2	DIG2	25CC	DIGIT	112D	DIGIT0	1144
DIGIT1	114C	DIGIT2	1151	DIGS	25DA	DIGS2	25DC
DLINE	2267	DLITE2	1E7A	DLITER	1E6A	DMINUS	143A
DMINX	1451	DO	247D	DOCOL	1525	DOCON	155F
DODOES	19F5	DOES	19E7	DOT	2634	DOTQ	1AB0
DOTQ1	1ACF	DOTQ2	1AD7	DOTR	25F1	DOUSER	1591
DOVAR	1579	DOVOC	1F0C	DP	1652	DPINIT	101E
DPL	16D2	DPLUS	1402	DPLUS2	1408	DRONE	2185
DROP	146C	DRZERO	2175	DTRAIL	1A62	DTRAL2	1A6A
DTRAL3	1A84	DTRAL4	1A88	DUP	1493	EDIGS	2584
ELSE	2525	EMIT	1221	ENCL2	11E2	ENCL3	11EF
ENCL4	11F4	ENCL5	1201	ENCL6	120A	ENCL7	1211
ENCL8	1215	ENCLOS	11D8	END	24D0	ENDIF	2457
EQUAL	1775	ERAM	2035	ERASE	1BF7	ERROR	1D6C
ERROR2	1D7A	EXEC	106F	EXPEC2	1B1F	EXPEC3	1B4D
EXPEC4	1B64	EXPEC5	1B66	EXPEC6	1B72	EXPECT	1B15
FENCE	1649	FENCIN	101C	FILL	1BD7	FIRST	15CF
FIX	283F	FLD	16DC	FORGET	23EB	FORTH	0150
FOUND	11B2	FROMR	13A6	GET	2812	GETX	102E
GO	2889	GREAT	179F	HERE	172C	HEX	197B
HI	2393	HLD	16F9	HOLD	1C17	I	111C
IDDOT	1D9D	IF	250E	IMMED	1ED4	IN	1674
IND2	27D1	INDENT	27C6	INDEX	268F	INDEX2	2699
INDEX3	26B1	INTER2	1E8A	INTER3	1EA2	INTER4	1EA6
INTER5	1EAA	INTER6	1EBE	INTER7	1EC2	INTERP	1E88
IOSTAT	0136	IP	00F2	KEY	1239	LATEST	183B
LBRAK	1948	LEAVE	137C	LESS	1781	LESSF	1791
LESST	1794	LESSX	1796	LFA	184B	LIMIT	15DB
LIST	264F	LIST2	266D	LIT	1049	LITER	1E4D
LITER2	1E5D	LO	238A	LOAD	22B2	LOOP	2490
MAX	17E7	MAX2	17F5	MEMEND	3210	MEMTOP	3FFF

MESS	227B	MESS3	229E	MESS4	22A9	MIN	17CF
MIN2	17DD	MINUS	1421	MINUS2	142C	MINUS3	142E
MOD	2085	MSMOD	20B8	MTBUF	2161	N	00E0
NBLK	0004	NEXT	1034	NEXT2	103A	NEXT3	103C
NFA	1868	NOOP	2747	NULL	1B9C	NULL1	1BC6
NULL2	1BCA	NULL3	1BCE	NUMB	1CDC	NUMB1	1CF9
NUMB2	1D1E	NUMB3	1D28	OFSET	1695	OK	2827
ONE	15AA	ONEP	1710	OR	1313	ORIG	1000
OUT	167E	OVER	145B	P4HEX	3F10	PA	00E4
PA0	00E2	PABORT	1D5E	PAD	1C2F	PAREN	1F32
PBUF	2113	PBUF2	2128	PC	00E6	PCR	233C
PCR2	234A	PD	00E0	PDOTQ	1A95	PEMIT	2300
PFA	187D	PFIND	1162	PFIND0	116B	PFIND1	1174
PFIND2	1187	PFIND3	119D	PFIND4	11A9	PFIND8	11A6
PFIND9	11AB	PKEY	2317	PLINE	2243	PLOOP	24A6
PLUS	13F1	PNUMB	1C91	PNUMB2	1C93	PNUMB3	1CC9
PNUMB4	1CCF	PORIG	1601	PQTER	232F	PQTER2	233B
PREV	2108	PROTO	3F80	PSCODE	19A5	PSTORE	14A1
PULABX	1026	PUSHBA	1032	QCOMP	18BE	QCSP	18FD
QDISC	235B	QERR	18A4	QERR2	18B2	QERR3	18B4
QEXEC	18D5	QLOAD	1919	QPAIRS	18EB	QSTAC2	1AF7
QSTAC3	1B0A	QSTACK	1AE2	QTERM	1250	QUERY	1B84
QUEST	2640	QUIT	1F42	QUIT2	1F4C	QUIT3	1F64
R	13B9	RAM	1FF2	RBRAK	1956	REND	0183
REPEAT	24F5	RFORTH	2002	RINIT	1014	RNUM	16EF
ROT	17AD	RP	00F4	RPSTOR	1358	RTASK	2031
RW	239D	RW2	23C0	RW3	23C8	RZERO	1619
SCR	1688	SCSP	1891	SEMI	153C	SEMIC	19BD
SEMIS	1367	SIGN	259D	SIGN2	25AC	SINIT	1012
SLASH	2075	SLMOD	2065	SMUDGE	196A	SPACE	17C1
SPACE2	2568	SPACE3	256E	SPACES	2558	SPAT	1337
SPSTOR	1348	SSLASH	20A6	SSMOD	2095	STABX	1028
STAR	2054	STATE	16BD	STOD	2046	STORE	14F0
SUB	1769	SWAP	147A	SZERO	1610	T2	2808
T3	2846	T4	2851	T4EX	2871	T4IN	2861
T5	2881	TASK	017F	THEN	2472	THREE	15BA
TIB	1623	TICK	23D2	TOGGLE	14BE	TOR	1390
TR0	276A	TR1	27D7	TR2	27DD	TRA	2765
TRACE	2759	TRACEM	00EB	TRAV	1817	TRAV2	181B
TREQL	277B	TREXIT	288C	TRIAD	26BF	TRIAD2	26D3
TRIAD3	26E1	TRLIM	00EA	TRNO	27F0	TRNO2	284F
TRP1	278D	TRP2	2789	TRP3	2785	TRYES	2796
TWO	15B2	TWOP	171D	TYPE	1A34	TYPE2	1A44
TYPE3	1A52	TYPE4	1A54	UNTIL	24BC	UORIG	0100
UP	00F6	UPDATE	213B	UPINIT	1010	USE	20FD
USER	158B	USL1	12D4	USL2	12DE	USL3	12E1
USL4	12E6	USL5	12F4	USLASH	12CA	USTAR	12A2
USTAR2	12B1	USTAR3	12BF	USTAR4	12C3	USTARS	12AB
VALAN	3F13	VAR	1573	VECT	00EE	VLIST	26F8
VLIST1	2707	VLIST2	2722	VOCAB	1EEC	VOCINT	1020
VOCLIN	1661	W	00F0	WAIT	27F2	WAIT2	283B
WAIT3	283D	WARM2	1FCC	WARN	163D	WENT	1FC6
WHILE	2547	WIDTH	162F	WORD	1C41	WORD2	1C55
WORD3	1C59	XBASE	0126	XBLK	0116	XCOLUM	0134
XCONT	0120	XCSP	012C	XCURR	0122	XDELAY	0132
XDO	10FF	XDP	0112	XDPL	0128	XFENCE	0110

XFLD	012A	XHLD	0130	XIN	0118	XLOOP	10BA
XOFSET	011E	XOR	1325	XOUT	011A	XPLOF	10E8
XPLONO	10F0	XPLOOP	10CB	XPLOP2	10CF	XPLOPS	10DD
XPREV	0142	XRNUM	012E	XRZERO	0108	XSCR	011C
XSPZER	0106	XSTATE	0124	XTIB	010A	XUSE	0140
XVOCL	0114	XWARN	010E	XWIDTH	010C	ZBNO	10A8
ZBRAN	108E	ZBYES	1097	ZEQU	13C7	ZEQU2	13D1
ZERO	15A2	ZLESS	13DA	ZLESS2	13E9	ZZZZ	2749